www.sciencemag.org/cgi/content/full/328/5975/208/DC1

Supporting Online Material for

Why Copy Others? Insights from the Social Learning Strategies Tournament

L. Rendell,* R. Boyd, D. Cownden, M. Enquist, K. Eriksson, M. W. Feldman, L. Fogarty, S. Ghirlanda, T. Lillicrap, K. N. Laland*

*To whom correspondence should be addressed. E-mail: ler4@st-andrews.ac.uk (L.R.); knl1@st-andrews.ac.uk (K.N.L.)

Published 9 April 2010, *Science* **328**, 208 (2010)
DOI: 10.1126/science.1184719

This PDF file includes:

Materials and Methods
SOM Text
Figs. S1 to S13
Tables S1 to S5
References
Appendices A to C

WHY COPY OTHERS? INSIGHTS FROM THE SOCIAL LEARNING STRATEGIES TOURNAMENT

Rendell, L., Boyd, R., Cownden, D., Enquist, M., Eriksson, K., Feldman, M. W., Fogarty, L., Ghirlanda, S., Lillicrap, T. & Laland, K. N.

SUPPLEMENTARY ONLINE MATERIAL

Methods

We held an open computer-based tournament to determine the most effective strategy for learning in a complex, changeable environment. To enter the tournament, applicants needed to devise a *strategy* – a set of rules that specified when an individual organism should perform each of the three moves in the game: (*i*) perform an established behavior from its repertoire (EXPLOIT), (*ii*) engage in trial-and-error learning (INNOVATE), or (*iii*) learn from other individuals (OBSERVE)[1].

Tournament simulation environment[2]

The simulation environment was represented as a 'restless multi-armed bandit' encompassing 100 possible behavioral acts (represented arbitrarily by the integer's 1-100) and a payoff associated with each act. The payoff for each act was an integer drawn at random from an exponential distribution (λ=1; values were squared, then doubled, and finally rounded to give integers mostly falling in the range 0-50). Payoffs changed between rounds with independent probability p_c, with new payoffs drawn at random from the same distribution. This information was kept deliberately vague to participants, so as to discourage overly specific solutions (see below).

[1] The full rules of entry for the tournament are given in Appendix A.

[2] MATLAB® *.m files of the tournament simulation engine and the winning strategy can be found at http://lalandlab.st-andrews.ac.uk/

Each simulation contained a population of 100 agents. Each agent possessed a behavioral repertoire, which was empty at the start of the agent's life. An agent's repertoire could subsequently only contain acts through some form of learning. In each round, agents could perform one of three possible moves, called INNOVATE, OBSERVE and EXPLOIT. These moves are summarised in Table S1, with further details below. The role of the entered strategies was to specify which of these three moves an agent should play in each simulation round, with optional reference to information, specified below, that was made available to that agent. Each agent was controlled by one of the entered strategies, assigned at the start of its life. Agents did not change strategy during their lives.

Table S1: Move available to agents in the tournament simulation

Move (represents)	**Information gained**	**Payoff gained**
INNOVATE (asocial learning)	Act and payoff (without error) randomly chosen from those currently unknown to the agent.	None.
OBSERVE (social learning)	Act and payoff of $n_{observe}$ demonstrators chosen at random from those playing EXPLOIT in the previous round. $N(0, \sigma_{payoffError})$ error always added to payoff information. Incorrect, randomly chosen, act returned with probability $p_{copyActWrong}$.	None.
EXPLOIT (performing a behavior)	Current actual payoff of chosen act.	Current actual payoff of chosen act.

Playing OBSERVE did not necessarily result in new behavior being learned. If no other agents played EXPLOIT in the last round, then nothing was learned. It was possible for an individual to OBSERVE an act already in its repertoire, in which case only the payoff recorded for that act was updated in the agent's behavioral repertoire. OBSERVE was error prone with regard to both act and payoff. OBSERVE returned a different act to that performed by the observed agent with probability $p_{copyActWrong}$, with the learned act selected at random from the 99 not being performed, although the payoff learned was still that of the observed agent. Independently, the returned payoff estimate was subject to normally distributed random error (rounded to the nearest integer) with mean 0 and standard deviation $\sigma_{payoffError}$ (with the returned payoff estimate lower bounded at 0). If an agent already had all 100 possible acts in its repertoire, it gained no new act from playing INNOVATE or OBSERVE and zeros were recorded in its history for that move.

An individual could only EXPLOIT behavioral acts it had previously learned. When an individual chose to EXPLOIT an act, it received the current payoff specified in the environment. Note that this value could differ from the expected payoff held in the agent's behavioral repertoire, for two reasons. Firstly, the payoff for an act could have changed in the rounds since it was learned or last exploited (with probability p_c each round). Secondly, if the act was learned in an OBSERVE move, then the payoff could have been subject to error. When an agent played EXPLOIT, we assumed it could, by performing an act, update its knowledge of how profitable that act was, and store the updated information in its behavioral repertoire.

We assumed agents could remember their own history of moves and payoffs, as well as their current behavioral repertoire. Along with the number of rounds the agent had been alive, this history and behavioral repertoire was the only information available to the entered strategies when deciding which move an agent should play.

Evolutionary dynamics

We modelled evolutionary change as a death-birth process. Within a simulation, agents died with probability 0.02 per round, giving an expected lifespan of 50 rounds. Dying individuals were replaced by the offspring of agents selected to reproduce from those surviving with probability proportional to the agent's mean lifetime payoff P. We defined an agent's P value to be the sum of all its payoffs from playing EXPLOIT during its life, divided by the number of rounds it had been alive. The probability of individual *z* reproducing was P_z / ΣP, where ΣP was the summed mean lifetime payoff of the population in that round.

Offspring usually carried their parent's strategy, except for a small probability of mutation, in which case the offspring carried one of the other strategies available in the simulation. While the mutation rate we used (0.02) is obviously high relative to natural rates of mutation in eukaryotes, we found that reducing this rate does not qualitatively affect our outcomes, and the higher rate offers significant computational advantages in terms of time to equilibrium.

Tournament structure rationale

It was clear from the outset that the scientific validity of the tournament would hinge critically of the precise details of the game. For this reason, the lead organizers of the tournament (Laland, Rendell) recruited a committee of leading authorities in the field of social learning research and game theory to advise them on tournament structure and design (Boyd, Enquist, Eriksson, Feldman). The primary function of this committee was to ensure, as far as possible, that the tournament was set up in a sensible way, such that it generated relevant insights into an empirically meaningful problem, one that could not easily be resolved through trivial solutions. The tournament structure went through several major design iterations, over a period of 18 months, and was tested extensively at each stage through simulation, including in an independent laboratory (Ghirlanda), and through mini tournaments. At the end of this validation the organizers and committee were entirely satisfied that the tournament presented a challenging and non-trivial problem.

We regarded it as important that the strategies be judged in a biologically meaningful context, such that effective strategies potentially shed light on social learning in humans and other animals. For instance, it was critical that strategies be evaluated in a spatially and temporally varying simulation environment, where multiple behavioural options were available. At the same time, it was important that the rules and specifications of the contest were simple enough to make interpretation of the outcome comprehensible, and to encourage the participation of entrants from a variety of different backgrounds. The chosen tournament structure reflects this compromise between validity and accessibility.

We considered a variety of possible structures to the tournament, including the tracking of a single environmental state, spatially-explicit grid environments, and multi-deme stepping-stone models drawn from population genetics. Our adoption of the 'restless multi-armed bandit' (i.e. a range of possible behaviors, each with its own, changeable, payoff) has the advantage both of being a well-understood problem independently of social learning research (*S1*), and currently intractable to analytical optimisation, despite considerable effort on the problem (e.g. *S2*). This structure therefore provided a familiar, valid, but non-trivial problem for the basis of the tournament.

Strategy evaluation Stage I: Round-robin pair-wise contests

Strategies first took part in pair-wise contests against all other strategies. Each pair-wise contest consisted of 10 simulations in which agents performing strategy A were introduced (using the mutation process described above) into a population containing only strategy B, and the reciprocal 10 simulations in which B was introduced into A-dominated populations. We adopted a reciprocal invasion approach to ensure our findings were robust to strategies' initial frequencies. In each simulation, a population of the dominant strategy was introduced and run for 100 rounds without mutation so that agents could establish their behavioral repertoires. Mutation was then introduced, providing the second strategy the opportunity to invade, and simulations were run for a further 10,000 rounds. The mean frequency of a strategy over the last 2,500 simulation rounds was its score for that simulation. *Simulation scores* were then averaged over the 20 simulations, and this average recorded as the *contest score* for that strategy in that contest. Strategies were then ranked on their average contest score once they had played against every other strategy. These simulations were run with the parameter set [p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05, $\sigma_{payoffError}$=1]. This stage involved 5,356 paired contests, with 107,120 (5,356×20) individual simulation runs.

Although we initially planned to simply carry the 10 highest scoring strategies through to the next tournament phase, we found that this division cut the payoff distribution within a series of strategies (ranked 6 to 24) with relatively small differences in their scores such that the initial ranking could plausibly have been dependent on the set of conditions we happened to choose (Main text, Figure 1a inset). Since we wanted to be confident that the apparent success of the 10 best strategies was not due a chance match of any strategy to the specific single parameter set, we elected to run more pair-wise contests on the top 24 strategies across a range of conditions.

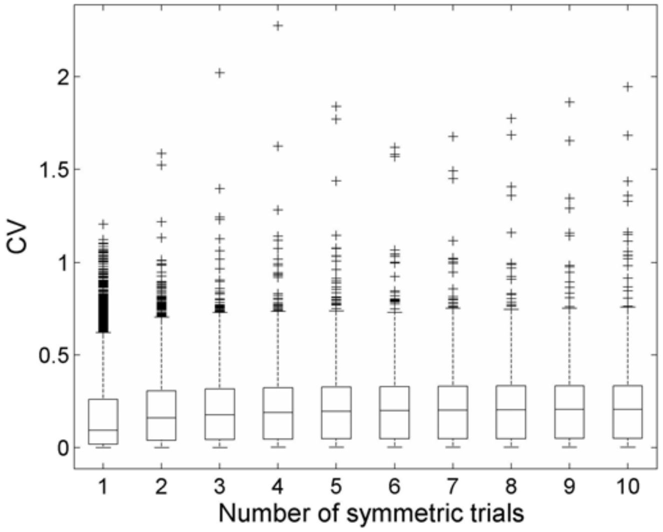
We found it was computationally feasible to run a further 8 conditions for the subset of strategies ranking in the top 24 of the first set of contests; these conditions are set out in Table S2. Note that we did not vary the parameter $\sigma_{copyPayoffError}$ in these conditions, as we reasoned that the parameter $p_{copyActWrong}$ would affect the accuracy of social learning in a similar manner but to a stronger degree; accordingly, to also vary $\sigma_{copyPayoffError}$ orthogonally would unnecessarily duplicate effort in exploring the effect of the accuracy of social learning as well as doubling the computation time required. We ran a single pair-wise run with two extra conditions, varying $\sigma_{copyPayoffError}$ from the initial condition above [p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05, $\sigma_{copyPayoffError}$=5], and [p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05, $\sigma_{copyPayoffError}$=10], but found that made no difference to the strategies that were eventually selected. For these further conditions we reduced the number of repetitions per contest to 3 symmetric repetitions (i.e. 3 runs with strategy A as invader and 3 runs with strategy B as invader) as opposed to the 10 such repetitions run for the initial pair-wise contest. We selected the value 3 based on Figure S1, which shows that the distribution of coefficient of variation values for each pair-wise contest does not change for more than 3 repetitions.

Table S2: Details of further conditions run for top 24 strategies

Condition	p_c	$n_{observe}$	$p_{copyActWrong}$	$\sigma_{copyPayoffError}$
1	0.001	1	0.01	1
2	0.1	1	0.01	1
3	0.001	1	0.1	1
4	0.1	1	0.1	1
5	0.001	6	0.01	1
6	0.1	6	0.01	1
7	0.001	6	0.1	1
8	0.1	6	0.1	1

These additional analyses required 13,248 further simulation runs (23×24×3×8). No strategy switched from the original pair-wise results by more than 11 places, and the average change in rank was 2.5 places, suggesting it was highly unlikely that any strategies outside the top 24 would have been elevated into the top 10. The extra conditions resulted in two strategies from the original best 10 (*senescence* and *observe3ThenExploit*) being dropped in favour of two others (*livingdog* and *valueVariance*) that had initially ranked 13 and 15 respectively.

Figure S1: Boxplot of pair-wise coefficient of variation distributions by number of repetitions.



Strategy evaluation Stage II: Melee contests

In each simulation, all ten of the strategies selected in Stage I competed simultaneously. Each simulation started with a population consisting purely of a simple strategy that did not use any social learning, but played INNOVATE on the first round of its life and subsequently played EXPLOIT continually with the single act that it acquired on the first round[3]. We used this strategy simply to avoid giving any of the ten competing strategies any advantage or disadvantage from being already established in the population. Mutation was introduced from round 1, providing the competing strategies with equal opportunity to invade. Simulations were run for 10,000 rounds, but mutation was turned off in the last quarter (i.e. rounds 7,500 – 10,000). The mean frequencies of each strategy over the last quarter of each run were recorded as the scores for each strategy in that simulation. Strategies were then ranked on their average score across all simulations.

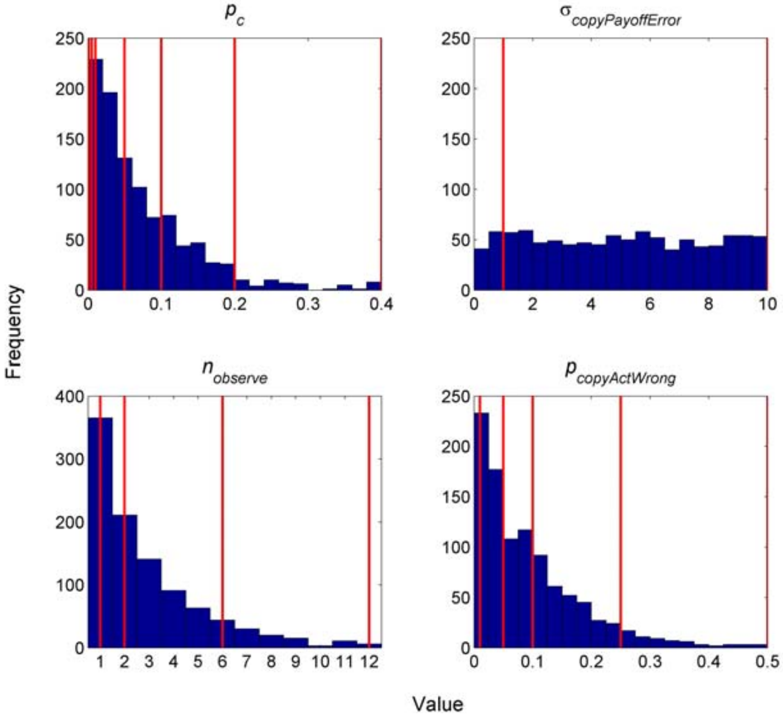
We ran two sets of conditions, which we termed *systematic* and *random*. For the systematic condition set, we selected a number of values for each of the four parameters, p_c, $n_{observe}$, $p_{copyActWrong}$, and $\sigma_{copyPayoffError}$ (Table S3). Fifty simulations were run with each of the 280 possible combinations of these parameter values giving 14,000 simulations. To check that the results of this process were not unduly affected by the specific parameter values we chose, we also ran random conditions, where parameter values were chosen at random from statistical distributions weighted in accordance with the values chosen for the systematic conditions (Figure S2). We weighted these distributions toward lower values of p_c, $n_{observe}$, and $p_{copyActWrong}$ because we considered higher values of these parameters to be less biologically or ecologically plausible than lower ones. We selected 1,000 unique sets of parameters values in this way and ran a single simulation with each set of values, giving a further 1,000 simulations. Systematic and random analyses gave identical returns on the ranked performance of the 10 strategies, computed across all simulations, based on their frequency in the last quarter of each simulation. Accordingly strategy scores were averaged over all 15,000 melee simulations to give the final scores.

[3] This strategy was entered independently in the tournament as *exploitOneInnovation*. It did not progress past the pairwise phase, ranking 102nd.

Table S3: Details of further conditions run for top 10 strategies; values in bold are those used for the main pair-wise contest in Stage I.

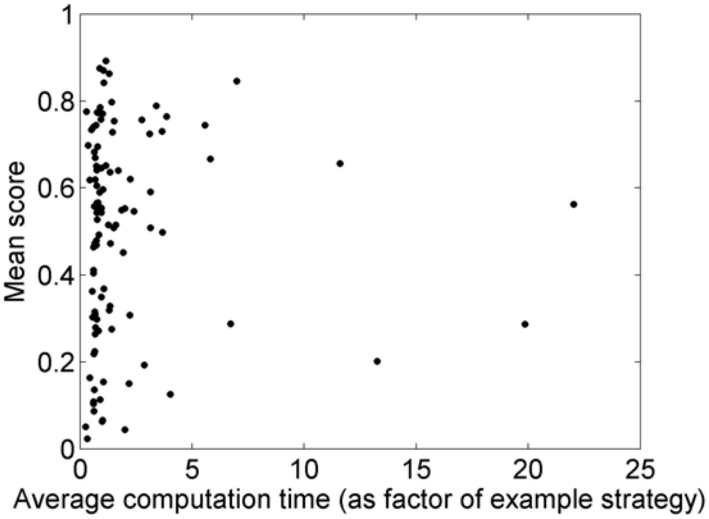
Parameter	Values						
p_c	0.001	0.005	**0.01**	0.05	0.1	0.2	0.4
$n_{observe}$	**1**	2	6	12			
$p_{copyActWrong}$	0.01	**0.05**	0.1	0.25	0.5		
$\sigma_{copyPayoffError}$	**1**	10					

Figure S2: Distributions of parameter values chosen for melee (Stage II). Blue bars are histograms of values chosen for the random conditions, red lines show values selected for systematic conditions.



All simulations were run in the Matlab®/Octave computing environment, using both the UK National Grid Service (*S3*) and desktop computers. Entries could take the form of Matlab®/Octave code or prose pseudocode; in the latter case, the pseudocode was converted to real code (by Rendell). We guarded against coding errors by having each strategy coded by a second independent coder (Fogarty), and testing that each version produced exactly identical results when given the same input sequences, including identical sequences of randomly-generated numbers when strategies made decisions stochastically. Strategies were restricted to take, on average, no longer than 25 times the duration of an example strategy provided in the rules to return a decision. No strategy failed this criterion, and there was no relationship at all between computation time and score in the pair-wise phase of the tournament (Figure S3).

Figure S3: Strategy scores in pair-wise tournament phase plotted against the average per-round computation time, expressed as a multiple of the time taken by an example strategy.



Information for entrants

The full rules of entry for the tournament are given in Appendix A. Entrants were not informed of the exact nature of the payoff distribution (see below) and on the exact values of four simulation parameters, although we did provide the possible ranges of the latter. We deliberately omitted these details so that contestants would be required to think in as general terms as possible in designing their strategies.

Statistical analyses

We examined the factors that made a strategy successful in the pair-wise-contests (Stage I) using linear multiple regression and model selection, with score as the dependent variable. For each strategy, we calculated a range of possible predictors of a strategy's score (Table S4), and entered these into an all-possible-subsets model comparison procedure. We ran analyses both with all strategies and only the top 24 to see whether the same factors responsible for success in the broadest context were also important when competing only against relatively successful strategies. In each case, we first used the package ‘leaps’ in the statistical package *R* to return the five best models for each subset of predictors, selected by Mallow’s Cp (*S4-6*). We then selected from that set the model that minimised AIC (*S7*), although results were very similar when model selection was based on BIC. Finally, we calculated predictor effect sizes as beta weights using the package ‘yhat’ (*S8*).

Table S4: Predictors entered into model selection

Predictor name	**Explanation**
Check central payoff? (Y=1, N=0)	Categorical variable indicating whether a strategy checked any central tendency (e.g. mean) in the payoff values in the agent's history, either from learning or EXPLOIT.
Check mean EXPLOIT? (Y=1, N=0)	Categorical variable indicating whether a strategy checked the mean payoff from playing EXPLOIT in the agent's history.
Estimate $n_{Observe}$? (Y=1, N=0)	Categorical variable indicating whether a strategy estimated the value of the parameter $n_{Observe}$.
Estimate p_c? (Y=1, N=0)	Categorical variable indicating whether a strategy estimated the rate of environmental change as given by the parameter p_c.
Flexible behavior? (Y=1, N=0)	Categorical variable indicating whether a strategy's choice of move was affected by the outcome of previous moves, or always followed a predetermined series of moves.
Log of variance in rounds to EXPLOIT	Pooled variance, across all agents with the strategy, in the number of rounds between the 'birth' of an agent with the strategy and the first time the agent played EXPLOIT (continuous measure). We took the log of this value as exploratory analysis showed a log-linear relationship with score.
Mean rounds between learning moves	Average number of rounds between any learning moves (OBSERVE or INNOVATE), across all agents with the strategy (continuous).
Proportion of learning moves	Average proportion of moves dedicated to learning (either OBSERVE or INNOVATE), across all agents with the strategy (continuous).
Proportion of learning that is OBSERVE	Average proportion of learning moves that were OBSERVE, across all agents with the strategy (continuous).
Stochastic? (Y=1, N=0)	Categorical variable indicating whether a strategy ever chose between actions stochastically, i.e. dependent on the draw of a random number.

Supplementary text: Detailed results

The tournament attracted 104 entries from a broad range of academic disciplines (Anthropology, Biology, Computer Science, Engineering, Environmental science, Ethology, Management, Mathematics, Neuroscience, Philosophy, Physics, Primatology, Psychology and Sociology) as well as outside of academia, and from 16 different countries (Belgium, Canada, Czech Republic, Denmark, Finland, France, Germany, Italy, Japan, Netherlands, Portugal, Spain, Sweden, Switzerland, UK and USA). A list of all submitted strategies and their placings in the tournament is found in Appendix B.

Stage I

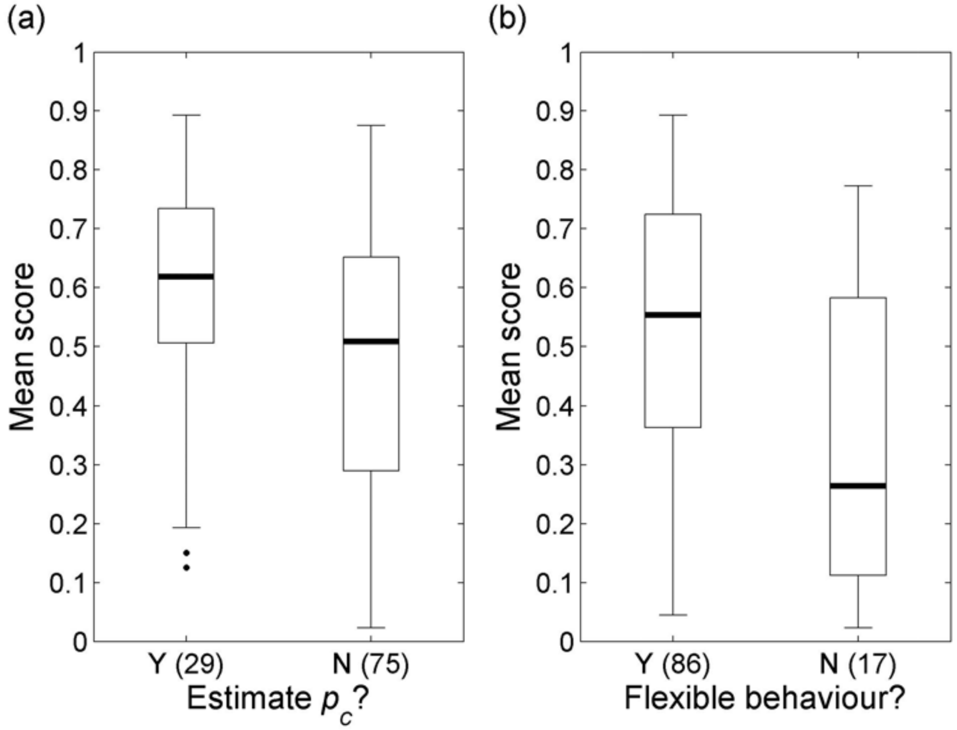
The initial pair-wise evaluation round involved 5,356 contests containing a total of 107,120 simulation runs. Strategy performance was unaffected by whether they were the invading or invaded strategy (Pearson correlation between invading and invaded scores, $r = 0.9998$, $p < 0.0001$).

In the statistical analysis including all strategies, there were 17 models within 3 AIC units of the best; no predictor present in the best model was absent in more than 6 of the 17, and no predictor absent in the best model was present in more than 8. When analysis was restricted to the best scoring 24 models, there were 12 models within 3 AIC units of the best; no predictor present in the best model was absent any of these models, and no predictor absent in the best was present in more than 3 of the other models, except for the categorical predictor specifying whether a strategy checked the mean payoff it had received from playing EXPLOIT in previous rounds. This predictor was retained in 7 of the 12 top models, but only 2 of the top 7. Fit diagnostics showed that the best models in both analyses had normally distributed and trend-free residuals, and both explained relatively large proportions of the data (adjusted $R^2 = 0.76$ and 0.50 respectively).

Statistical analysis of all 104 strategies returned a best model containing 5 predictors of a strategy's score, although not all were significant at $\alpha = 0.05$ in that model (Main text, Table 1).

We found strong effects of “Proportion of learning that is OBSERVE” and “Variance in time to first EXPLOIT”, and moderate effects of “Proportion of learning moves” and “Average rounds between learning moves”, which are discussed in the main text. The categorical variable indicating whether a strategy estimated the rate of environmental change apparently had a positive effect, but high variability within categories meant that the mean effects were not significant at $\alpha = 0.05$ (Figure S5a).

Figure S5: Box plots showing scores for strategies that did or did not (*a*) estimate the rate of environmental change and (*b*) have flexible behavior in the sense that behavior was conditional on the move history or current repertoire of an agent. Data are from pair-wise contests.



When the same analysis was restricted to just the 24 top-scoring strategies a different, and reduced, set of predictors emerged (Table S5). The best fit model in this analysis was not able to explain as much variation as the analysis with all strategies.

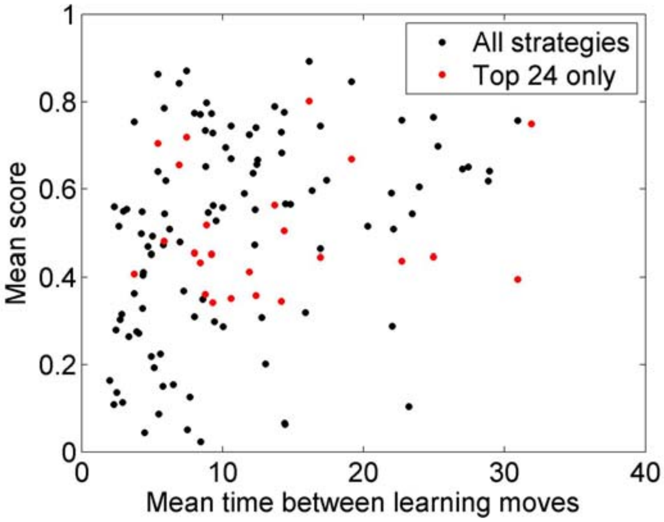
Table S5: Best model for pair-wise strategy scores using top 24 strategies only. Adjusted R^2=0.50.

Predictor	**Effect size (*β* weight)**	***β***	**S.E.**	***t***	***p*(>|*t*|)**
(Intercept)	-	0.98	0.14	6.95	<0.0001
Flexible behavior? (Y=1, N=0)	0.69	0.34	0.13	2.75	0.0127
Proportion of learning moves	-0.64	-2.74	0.78	-3.52	0.0023
Average rounds between learning moves	-0.61	-0.01	0.01	-2.35	0.0298
Variance in rounds to first EXPLOIT*	-0.47	-0.05	0.02	-3.11	0.0058

*We used the natural log of this predictor to give a better linear relationship

The categorical variable indicating whether a strategy had flexible behavior was retained with the largest effect size, in place of the variable indicating whether a strategy estimated the value of p_c (Figure S5b). The proportion of learning moves was retained with a large negative effect. The log of the variance in time to EXPLOIT was also retained with a significant negative effect, as in the model with all strategies. Two predictors present in the model with all strategies were dropped in this model – the proportion of learning moves dedicated to OBSERVE, and the categorical variable indicating whether a strategy estimated the rate of environmental change. Finally, the mean number of rounds between learning moves had a significant effect in both analyses. However, the effect is in opposite directions when considering data from all strategies, where there is an apparent positive relationship, compared to data from only the top 24 strategies, where the effect is negative (Figure S6). Thus, when competing against other effective strategies, it was detrimental to leave too many rounds between learning moves.

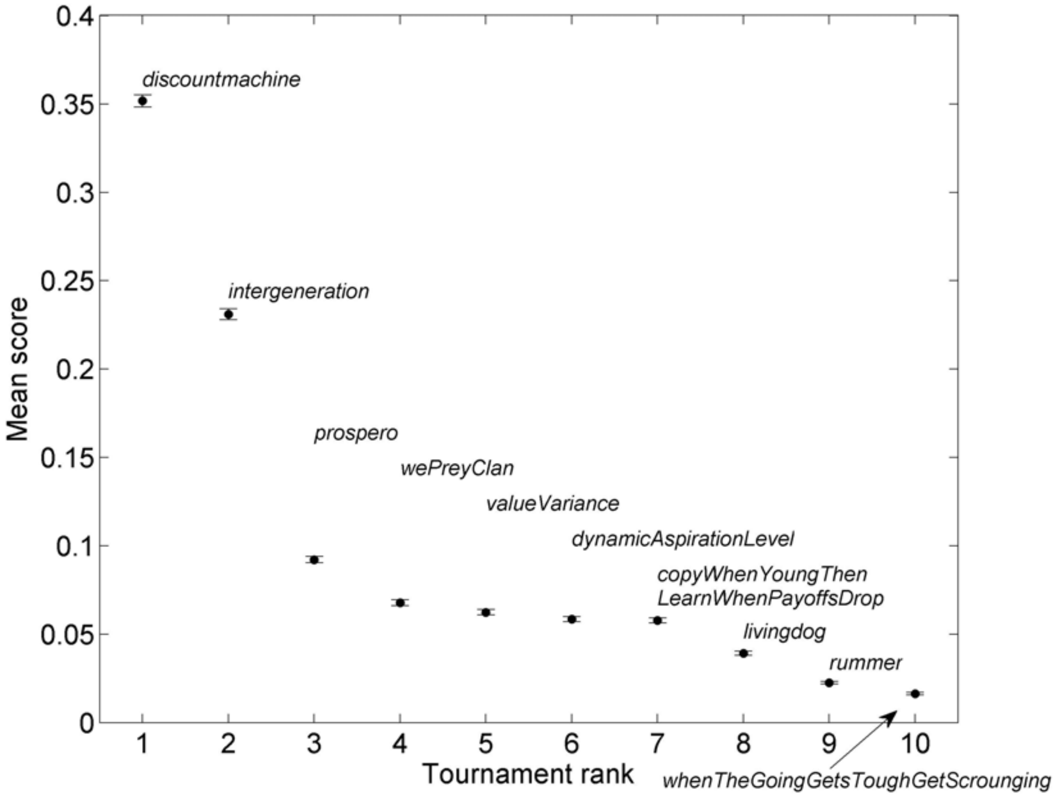
Figure S6: Final score against the mean number of rounds between learning moves. Data from all pair-wise contests are labelled 'All strategies', data restricted to contests between the top 24 strategies are labelled 'Top 24 only'. Note that for the latter, the mean score is calculated from contests involving only those 24 strategies, so appear lower than might initially be expected.



Stage II analysis

The ten highest scoring strategies from the pair-wise phase then progressed to the melee phase, in which all ten strategies competed simultaneously in series of simulations across a broad range of parameter values. (Descriptions of the top 10 strategies can be found in Appendix C). Strategies were ranked according to their score averaged over all melee simulations. The highest scoring strategy in this phase, and therefore the tournament winner, *discountmachine*, was the same strategy that scored highest in the first, pair-wise, phase, and scored highest in both the random and systematic analysis of the melee. This strategy won convincingly, although with the second placed strategy, *intergeneration*, it formed a pair of strategies that performed markedly better than the other contenders (Figure S7).

Figure S7: Ranked overall strategy scores from melee contests, incorporating random and systematic melee conditions. Error bars are ± SEM, although not always visible as all SEMs<0.004



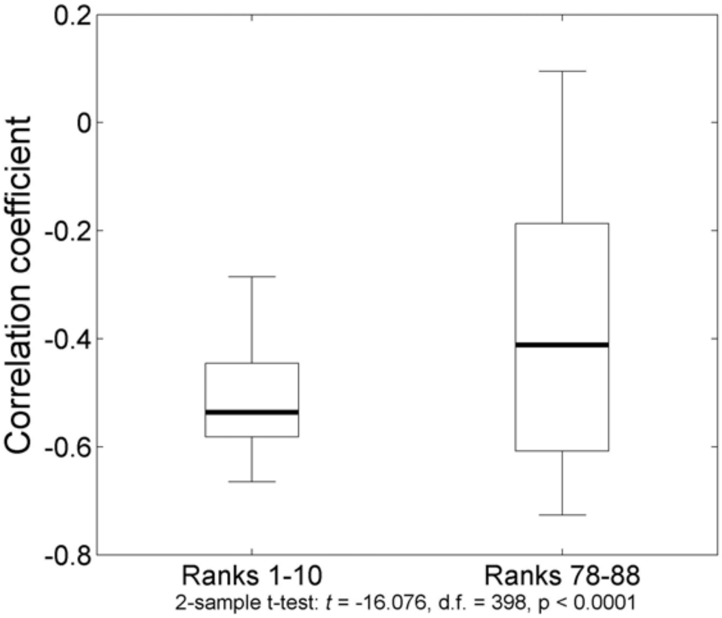
We found that all ten melee strategies were responsive to changes in the rate of environmental variation in that they all increased the amount of learning they did at higher rates of variation (Main text Figure 3a-b). Most strategies continued to increase the amount of learning as variation rates increased, although four did not, including the top two, in that they appeared to cap the amount of learning they did even as rates of environmental change continued to increase. The second placed strategy stands out has having the lowest learning rates of all the melee contenders. While all strategies continued to learn to some extent throughout the agent's lives, the winner stood out by distributing learning almost equally across different phases of life (Main text Figure 3b). In contrast, the second placed strategy had the highest variance in learning rates, concentrating over 60% of its learning in the first third of the agent's lives. This contrast goes some way to explaining the relative performance of the strategies in varying rates of environmental change. When the environment is relatively static, low learning rates overall and a concentration at the beginning of life are advantageous to *intergeneration*, as the acquired information is less likely to change, while in changeable environments, the higher lifelong learning rates of *discountmachine* give it the upper hand.

When learning, all melee strategies used OBSERVE at least 50% of the time, regardless of the conditions (Main text Figure 3c-d). There was a great deal of variation in how strategies changed their use of social and asocial learning as conditions varied. Notably, the top two ranked strategies, as well as two others (*wePreyClan* and *dynamicAspirationLevel*, ranked 4th and 6th respectively) played OBSERVE almost all the time, regardless of how much the environment was changing (Main text Figure 3c) or what the relative costs of social and asocial learning were (Main text Figure 3d). The other strategies showed a variety of responses to both variables, with some increasing the amount of social learning with increasing environmental variation and reduced cost of social learning, and others decreasing the amount of social learning under the same conditions.

In general, successful strategies were able to target the timing of their learning moves effectively, increasing the amount of learning in periods immediately following significant drops in average lifetime payoff in the population caused by environmental change that reduced the payoff of a commonly exploited act, but also quickly dropping back to low levels of learning so as to maximize the amount of exploiting (Main text Figure 2c). To quantify this, we calculated the maximum absolute lagged Pearson correlation value between the time series of the average lifetime payoff in the population and the proportion of the population playing a learning move, for 200 of the random melee simulations. To compare with less effective strategies, we selected the strategy *piRounds*, that chose an action based on the digit of π that corresponded to the age of an agent (i.e. behaved at random) and which ranked 88 in the pair-wise phase, and the nine strategies ranked immediately above it, and calculated the same maximum correlation values for these strategies when they played 200 rounds under the same conditions.

For melee strategies, the largest absolute correlations were always negative (Figure S8), and always with a positive lag of 1 or 2, indicating a rapid increase in learning almost immediately after payoff drops. In contrast, the maximum correlations for the less effective strategies were less strong, not always negative and occurred at a more diverse range of lags.

Figure S8: Boxplot of maximum absolute lagged Pearson correlation values between average lifetime payoff and proportion of learning in a population for effective (Rank 1-10) and less effective (Rank 78-88) strategies.

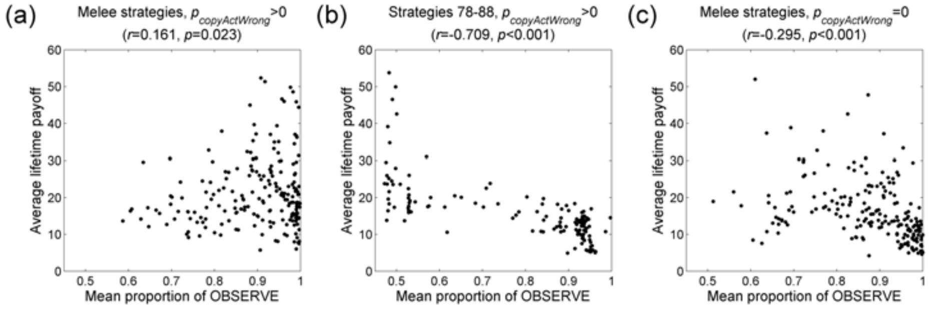


The timing of learning was not, however, the only key to success. The winning strategy used social learning virtually exclusively – it would play INNOVATE only on the second round of an agent's life if, after playing OBSERVE on the first, there was no behavior observed, i.e. no other agent in the population played EXPLOIT. That this reliance on social learning was crucial to its success is shown by the results of running the random conditions melee again but with a version of *discountmachine* re-coded to learn only by playing INNOVATE (Main text Figure 1b). Note that for this and subsequent analyses we compared scores only in the *random* conditions segment of the melee round, which is why the scores in Figures S7 and Figure 1b are different – the former shows the score for the entire melee stage, the latter only for the random conditions segment. We did this for reasons of computation time, as the *random* conditions treatment only require 1,000 simulations, so we could perform multiple analyses in a reasonable time, while also making sure a reasonable proportion of the parameter space was covered. In this analysis, the innovate-only version places last against the other melee strategies, and, interestingly, other strategies change scores significantly, such that the second placed tournament strategy does not win and is instead overtaken by 4 other strategies. This result suggests that there are frequency-dependent effects present. Seemingly, *discountmachine* inhibits the fitness of other melee strategies when it relies exclusively on OBSERVE.

Existing theory has suggested that high rates of social learning in a population can be detrimental to the average fitness of individuals in that population (*S9-10*). This would appear not to be the case with the melee strategies in the tournament (Figure S9a). We also looked at this relationship for strategies that had performed relatively poorly in the first, pair-wise, phase of the tournament, running 200 random condition melee rounds with the strategies that ranked 78-88 for comparison. We found that for poorly performing strategies the relationship between average individual fitness and the rate of social learning was strongly negative (Figure S9b), the complete opposite of the result for the melee strategies.

This contrast between our results and previous theory can be explained by noting that the tournament structure contained a mechanism by which social learning can result in new behavior entering the population, through the parameter $p_{copyActWrong}$, the probability that OBSERVE returns not the observed act, but another randomly selected act. When we ran 200 random condition melee rounds with the melee strategies but with $p_{copyActWrong}$ set to zero, the positive correlation we found between average individual fitness and the rate of social learning amongst melee strategies disappears and becomes instead strongly negative (r = -0.30, p < 0.001; Figure S9c). Thus, when there is no copy error, high levels of social learning are associated with reduced average individual fitness in the population.

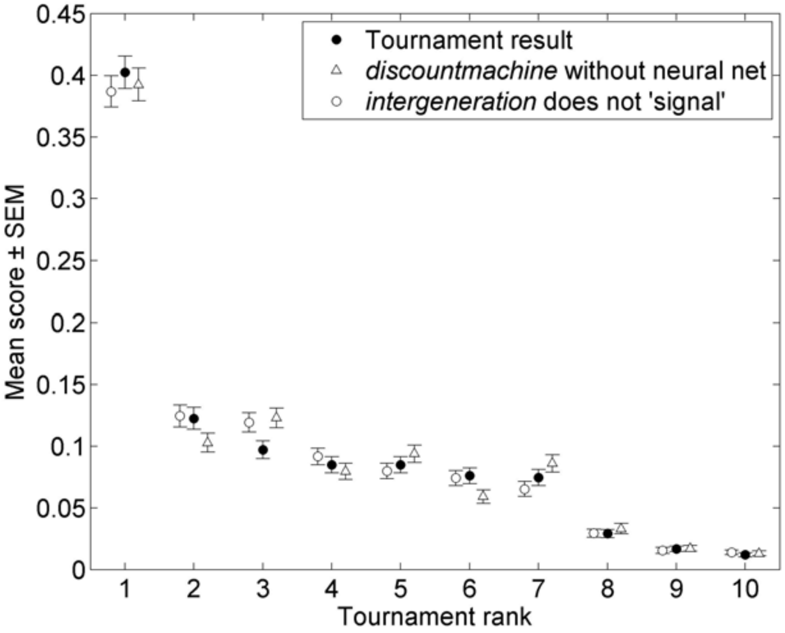
Figure S9: The average lifetime payoff in a population against the mean proportion of OBSERVE when learning, for (a) strategies in melee phase with $p_{copyActWrong} > 0$, (b) strategies ranked 78-88 in pair-wise phase with $p_{copyActWrong} > 0$, and (c) strategies in melee phase with $p_{copyActWrong}$ fixed at 0. Results are means over the last quarter of 200 simulations across randomly selected conditions.



The effect of $p_{copyActWrong}$ and the presence of frequency dependent effects is further illustrated by analysis of the performance of each strategy by itself. For all melee strategies, we ran single simulations containing only one strategy, using the same conditions as in the pair-wise tournament phase. We then ran the same simulations again but with $p_{copyActWrong}$ set to zero, and compared results in terms of the average individual mean lifetime payoff in each population. Under the pair-wise conditions, we found a strong inverse relationship between the mean lifetime payoffs of strategies playing alone and their scores in the tournament melee – lower ranked strategies had higher fitness when playing alone than those ranked higher (Main text Figure 1d). The effect of setting $p_{copyActWrong} = 0$ is dramatic for those strategies that rely exclusively on OBSERVE, with the average individual payoffs in populations containing only those strategies dropping to one quarter or less of their previous values. This again suggests that copy error is a significant source of novel behaviour. However, the strategy that ranked 6th, *dynamicAspirationLevel*, while relying heavily on OBSERVE, did not do so exclusively (average proportion of learning moves that were OBSERVE was 0.995 across all melee simulations), and its performance when playing by itself was unaffected by setting $p_{copyActWrong} = 0$; thus in our model relatively small amounts of innovation can bring in enough new behavior to maintain payoff levels.

The two most successful strategies, *discountmachine* and *intergeneration*, each had unique features: the former had a neural network, which it used to decide between learning and exploiting alternatives, while the latter deployed behavior designed to pass signals from older to younger agents regarding what should be considered a good payoff. Our analyses suggest, however, that it was not these unique features that were crucial to their success, as re-runs of the *random* conditions melee with versions of these strategies coded to remove these unique features produced results identical to the original tournament (Figure S10). We further investigated the role of the neural network in the success of the winner, *discountmachine*, by playing it against the version of itself without a neural network across 1,000 random melee conditions. We found that the complete version tended to do increasingly better than the reduced version as p_c increased (linear regression of difference between the scores of the strategies against p_c across 1,000 conditions: $\beta = 1.78$, s.e. = 0.2, $t = 8.97$, d.f. = 998, $p < 0.00001$), indicating that under certain conditions the neural network did make a positive contribution to the strategy's performance.

Figure S10: Ranked scores from tournament random conditions melee and from runs under the same conditions with the strategies *discountmachine* and *intergeneration* re-coded to remove, respectively, the neural network in *discountmachine* and the attempted intergenerational signalling of *intergeneration*.



The winning strategy, *discountmachine*, also adopted a forward-looking approach to making decisions in a way not seen in any of the other strategies. It chose between EXPLOIT and OBSERVE by using the closed form of a geometric series to compare the expected payoff gains from each move. It considered the gains expected from either exploiting the best act currently known until death or a change in payoff, or observing once and then exploiting the expected observed payoff, again until death or a change in payoff. The strategy chose to play OBSERVE if

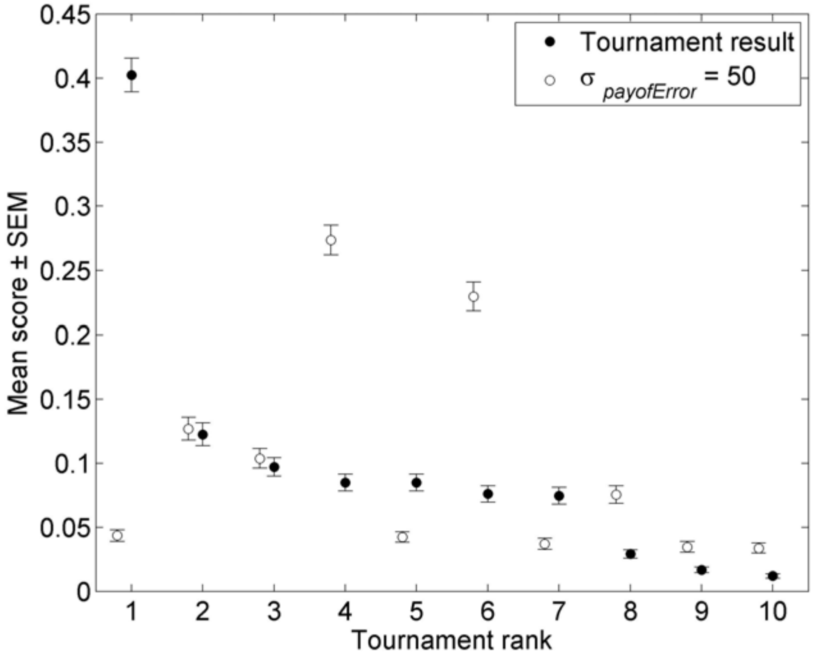
$$w_{max}\left(\frac{1}{1-d}\right) < O_{est}\left(\frac{d}{1-d}\right), \qquad (2)$$

where w_{max} is the maximum of the expected payoffs currently available in the agent's repertoire, O_{est} is an estimate of the expected payoff of an observed act calculated simply as the mean of all the observed payoffs in the agent's history, and *d* is a 'discounting' factor given by the product of the probability the agent will be alive in the next iteration and the agent's current estimate of p_c, or

$$d = (1 - p_{est})(1 - p_{death}). \tag{3}$$

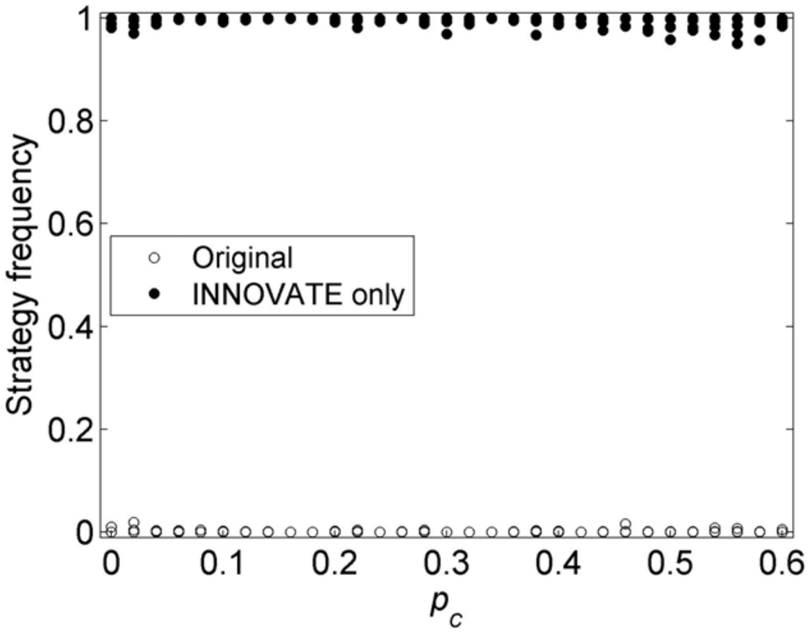
In a further analysis, we wanted to explore how robust our findings were to the assumption that OBSERVE revealed information about the payoff of a behaviour as well as the behaviour itself. To do this we ran the random conditions melee a further time, devaluing information about social learning payoffs by making payoff observation extremely unreliable (setting sigma, the standard deviation of payoff observation error, to 50, when payoffs themselves are generally in the range 0-50). Under these conditions OBSERVE essentially provides no information about payoff. Nonetheless, while the tournament result is altered in the sense that a different winner emerges, the new winning and second-place strategies, which ranked 4th and 6th in the tournament proper, also use social learning in >95% and >97% of all learning moves respectively (Figure S11). Thus the success of social learning in the tournament does not depend on the ability to observe demonstrator payoffs.

Figure S11: Ranked scores from tournament random conditions melee and from runs under the same conditions except with $\sigma_{payoffError}$ set to 50 so as to make the OBSERVE move uninformative with regard to the payoffs of the learned behaviour.



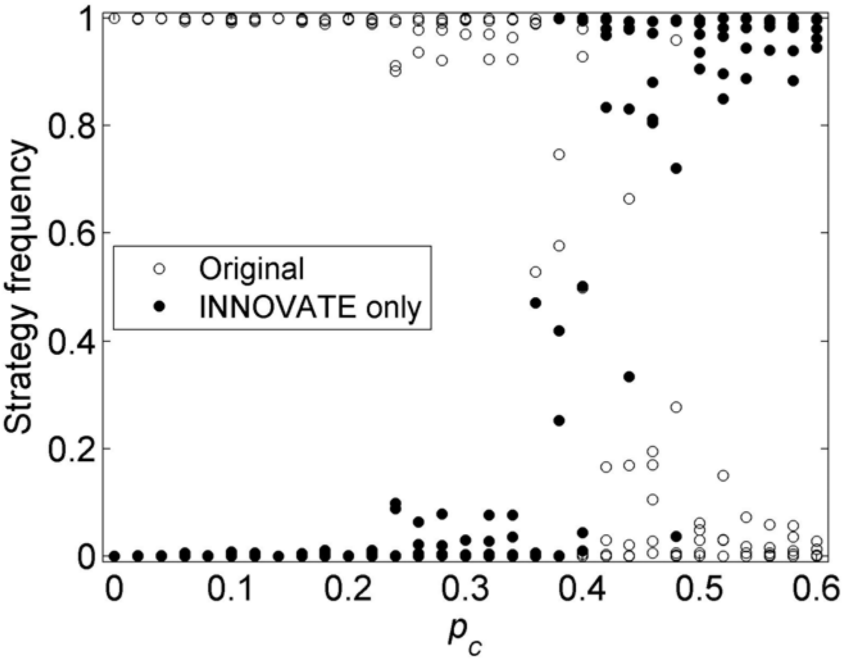
We also explored the extent to which the filtering of adaptive information by demonstrators underpinned the success of social learning in the tournament. We did this by running an alternative simulation model in which OBSERVE returned a behavior chosen at random from a demonstrator's repertoire with the behaviour the demonstrator had chosen to exploit removed, thereby preventing the filtering of information by rational agents choosing to exploit their best behaviour. We ran a series of such modified simulations in which the tournament winner and a version of itself altered to learn only by INNOVATE played against each other, together with the *exploitOneInnovation* strategy used to initiate simulations in the melee phase of the tournament. We systematically varied the rate of environmental change (p_c) across simulations. Five simulations were run at each level of p_c, and the other parameters were fixed at $n_{observe}$=1, $p_{copyActWrong}$=0.05, and $\sigma_{payoffError}$=1, identical to the first phase of the tournament. The results showed that, in contrast to Figure 5 (main text), *discountmachine*'s innovating cousin generally dominated the population irrespective of the rate of environmental change (Figure S12). A second analysis, using simulations in which OBSERVE returned a behavior chosen at random from a demonstrator's repertoire with the behaviour the demonstrator had chosen to exploit retained, also dramatically reduced the range over which social learning prospered, restricting this to highly stable environmental conditions. These results clearly demonstrate that the filtering of information by informed individuals is crucial to the success of social learning. In the absence of this filtering, social learning is in fact costly enough, through its associated errors and propensity to fail to introduce new behaviour to an agent (which occurred at a rate of 53% of OBSERVE moves in the first phase of the tournament), to be selectively disadvantaged.

Figure S12: Results of a series of simulations in which the tournament winner played against a version of itself altered to learn only by INNOVATE in a model where OBSERVE returned a behaviour selected at random from a demonstrator's repertoire. Five simulations were run at each level of *p_c*.



We are confident that it is this information filtering which underpins the success of social learning, and not our assumption that copying errors always return some behaviour, because of the results of a further set of simulations that we ran mirroring those described above. In this case, the OBSERVE move functioned as in the original tournament – it returned the behaviour being exploited by the demonstrator, so that filtering could work – except that in the event of a behavioural copying error, it returned nothing, rather than a behaviour randomly selected from those not observed. We again pitted *discountmachine* against its innovating cousin across a range of p_c values, fixing the parameter $p_{copyActWrong}$ at 0.05, as it had been in the first tournament phase. The result showed almost no difference compared to those obtained with copying error returning a random behaviour (Figure S13, compared to Figure 5 in the main text). These results demonstrate how robust our principal findings are to changes in the assumptions underlying our simulation model.

Figure S13: Results of a series of simulations in which the tournament winner played against a version of itself altered to learn only by INNOVATE in a model where OBSERVE returned no behaviour in the event of a copying error (rather than a randomly selected behaviour as in the original tournament). Five simulations were run at each level of p_c.



SOM References

S1. P. Whittle, *J. Appl. Probab.* 25, 287 (1988).

S2. C. H. Papadimitriou, J. N. Tsitsiklis, *Math. Oper. Res.* 24, 293 (1999).

S3. A. Richards, G. M. Sinclair, in *Grid Computing: Infrastructure, Service, and Applications,* L. Wang, J. Chen, W. Jie, Eds. (CRC Press, Boca Raton, FL, 2009).

S4. A. Miller, *Subset Selection in Regression*. (CRC Press, Boca Raton, FL., 2002).

S5. T. Lumley, *Package 'leaps' using Fortran code by Alan Miller.* http://cran.r-project.org/web/packages/leaps/index.html, (2009).

S6. R Development Core Team, *R: A language and environment for statistical computing*. (R Foundation for Statistical Computing, Vienna, Austria, 2004).

S7. K. P. Burnham, D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. (Springer, New York, ed. 2nd, 2002).

S8. K. Nimon, M. Lewis, R. Kane, R. M. Hayes, *Behavior Research Methods* 40, 457 (2008).

S9. R. Boyd, P. J. Richerson, *Culture and the Evolutionary Process*. (Chicago University Press, Chicago, 1985).

S10. A. Rogers, *Am. Anthropol.* 90, 819 (1988).

S11. R. Axelrod, *The Evolution of Cooperation*. (Basic Books, New York, 1984).

S12. R. Boyd, P. J. Richerson, *Ethol. Sociobiol.* 16, 123 (1995).

S13. L.-A. Giraldeau, T. J. Valone, J. J. Templeton, *Philos. Trans. R. Soc. Lond. (B Biol. Sci.)* 357, 1559 (2003).

S14. C. J. Barnard, R. M. Sibly, *Anim. Behav.* 29, 543 (1981).

S15. M. W. Feldman, K. Aoki, J. Kumm, *Anthropol. Sci.* 104, 209 (1996).

S16. M. Enquist, K. Eriksson, S. Ghirlanda, *Am. Anthropol.* 109, 727 (2007).

S17. B. G. Galef Jr., *Anim. Behav.* 49, 1325 (1995).

S18. J. Henrich, R. McElreath, *Evol. Anthropol.* 12, 123 (2003).

S19. K. N. Laland, *Learn. Behav.* 32, 4 (2004).

S20. K. H. Schlag, *J. Econ. Theory* 78, 130 (1998).

Appendix A: Tournament rules of entry

Social Learning Strategies Tournament

Rules for entry

Kevin Laland and Luke Rendell, University of St Andrews
4th January 2008

We are holding an open, computer-based, tournament to determine the most effective social learning strategy or strategies, as part of the EU 'Cultaptation' project. The author(s) of the winning entry will receive a cash prize of €10,000. This document contains background information on the rationale for the tournament, a description of how the tournament will be run, and technical details of how to enter. It is designed to provide all the necessary information needed by anyone wishing to enter the tournament. The primary contact for all issues regarding entries and any queries not covered here is Luke Rendell (ler4@st-andrews.ac.uk).

"This tournament is a wonderful opportunity to advance our understanding of the evolution of social learning, and I was glad to have been able to give advice about the rules. It has my wholehearted support and I hope that as many people as possible will have a go."

Robert Axelrod, University of Michigan

THE TOURNAMENT: BACKGROUND AND OBJECTIVES

Introduction

In recent years there has been growing interest (spanning several research fields, but especially economics, anthropology and biology), in the problem of how best to acquire valuable information from others. Mathematical and computational solutions to this problem are starting to emerge, often using game-theoretical approaches. We judge that the time is now right for a tournament, inspired by Robert Axelrod's famous Prisoner's Dilemma tournament on the evolution of cooperation (*S11*), but with the objective of establishing the most effective strategies for learning from others. We have received funding to organize such a tournament from the European Commission as part of the EU-NEST 'Cultaptation' project (www.intercult.su.se/cultaptation/). We hope that the competition will increase understanding of, and stimulate research on, social learning strategies, as Axelrod's tournament did for research on the evolution of cooperation.

Background

It is commonly assumed that social learning is inherently worthwhile. Individuals are deemed to benefit by copying because they take a short cut to acquiring adaptive information, saving themselves the costs of asocial (e.g. trial-and-error) learning. Copying, it is assumed, has the advantage that individuals do not need to re-invent technology, devise novel solutions, or evaluate environments for themselves. Intuitive though this argument may be, it is flawed (*S9-10, 12-13*). Copying others *per se* is not a recipe for success. This is easy to understand if social learning is regarded as a form of parasitism on information (*S13*): asocial learners are information *producers*, while social learners are information *scroungers*. Game-theoretical models of producer-scrounger interactions reveal that scroungers do better than producers only when fellow scroungers are rare, while at equilibrium their payoffs are equal (*S14*). Similarly, theoretical analyses of the evolution of social learning in a changing environment (*Se.g. 9, 10, 12, 15*) reveal that social learners have higher fitness than asocial learners when copying is rare, because most 'demonstrators' are asocial learners who will have sampled accurate information about the environment at some cost. As the frequency of social learning increases, the value of copying declines, because the proportion of asocial learners producing reliable information appropriate to the observer is decreasing. An equilibrium is reached with a mixture of social and asocial learning (*S16*). These mathematical analyses, together with more conceptual theory (*S*e.g. *17*), imply that copying others indiscriminately is not adaptive; rather, individuals must use social learning *selectively*, and learn asocially some of the time. Natural selection in animals capable of social learning ought to have fashioned specific adaptive *social learning strategies* that dictate the circumstances under which individuals will exploit information provided by others (*S9, 18-20*). At present, it is not clear which social learning strategy, if any, is best. The tournament has been set up to address this question.

Objective for entrants

To enter the tournament, you need to devise a *strategy* – a set of rules that specify when an individual organism should perform an established behaviour from its repertoire (EXPLOIT), when it should engage in trial-and-error learning (INNOVATE) and when it should learn from other individuals (OBSERVE) in deciding how to behave in a spatially and/or temporally variable environment. Performing the right behaviour is important, as fitness depends on how well behaviour is matched to the current environment. However, learning is not free, and fitness costs are imposed each time an individual learns for itself, or samples the behaviour of other individuals in its environment. For the purposes of the tournament, organisms will be assumed to know their own individual histories of behaviour and the fitness payoffs they received.

Strategies will be tested in a computational simulation framework. The specification of the simulations, details on how to enter, and detailed tournament rules are given in the technical details section below. Entrants should ensure they are familiar with this material, as the details given are crucial in ensuring that your strategy will be considered in the tournament.

Strategy evaluation

Strategies will take part in a two-stage competition; this is summarised here and full details are provided in section 2 below.

Stage 1: Strategies will take part in round-robin contests between all pairs of entered strategies. A contest, say between strategies A and B, involves exploring whether strategy A can invade a population containing only strategy B, and vice-versa. Each contest will involve several repeated simulations, with each strategy as the invader 50% of the time.

In each simulation, after a fixed number of iterations, the frequency of each strategy (that is, the proportion of the population with that strategy) will be recorded, and the average frequency across repetitions will be the score of that strategy in that contest.

Stage 2: At least the first ten highest scoring strategies will then be entered into a *melee* in which all strategies compete simultaneously in a range of simulation conditions. After a fixed number of rounds, the frequency of each strategy will be the score for that strategy. The procedure will be repeated and the strategy with the highest average score deemed the winner. Participants should therefore try and construct their strategies so that they are likely to work well under most conditions.

Committee

The tournament will be organised and run by Kevin Laland and Luke Rendell, both of the University of St Andrews. A committee has been formed to oversee the running of the tournament, and formally adjudicate when necessary, to ensure that the contest is run transparently. The committee is composed of the organizers plus the following persons, all of whom have expertise relevant to the tournament:

Robert Boyd, University of California, Los Angeles
Magnus Enquist, University of Stockholm
Kimmo Eriksson, Mälardalen University
Marcus Feldman, Stanford University

This committee has been extensively involved in designing of the tournament; we are also very grateful to Robert Axelrod of the University of Michigan for providing important advice and support with regard to the tournament design.

TECHNICAL DETAILS

1. Simulation specifications

Each simulation will contain a population of 100 individuals, and run for up to 10,000 rounds[4]. A single round will consist of the following computational steps:

> (i) Individuals are selected sequentially to choose a move (see below) until all individuals have played.
> (ii) Individuals reproduce with probabilities proportional to their average lifetime payoffs.
> (iii) The environment changes.

1.1 Environment and behaviour

1.1.1 The environment will be represented as a 'multi-arm bandit' wherein actors select from a range of possible behavioural acts and receive a payoff associated with that act. There will be 100 possible acts, and the payoff for each act will be chosen at the start of each simulation from a distribution with many relatively small payoffs and some

[4] If it is found that results are identical for shorter simulation runs then we may reduce this number for computational convenience.

relatively large ones. Therefore the environment can be represented a table with two rows associating behavioural acts with payoffs, for example:

Act:	1	2	3	4	5	…	100
Payoff:	4	0	17	1	7	…	3

1.1.2 The environment is not constant, and the payoff associated with a given behavioural act will change between each round of the simulation with a fixed probability, $p_{c.}$ There is no association between payoffs for acts before and after the environment changes - the new payoff will be chosen at random (from the same distribution used in 1.1.1). The payoff for each act will change independently of the others, so that p_c also represents the average proportion of payoffs that change in each round. In the round-robin stage of the tournament, p_c will initially be fixed to a single value drawn from the range between 0.001 and 0.4, but we may test multiple levels if computational constraints permit. In the *melee* stage, we will run simulations with varying levels of p_c, drawn from the range [0.001-0.4].

1.1.3 The simulations will contain a single population of 100 individuals, representing a focal deme embedded in a meta-population. Each individual will have a behavioural repertoire, containing a subset of the acts from the table specified above. Individuals are born naïve; they have an empty repertoire. Each individual's repertoire can subsequently contain only those acts, and knowledge about their payoffs, that are acquired through some form of learning (see below). Note that environmental change means that the payoff recorded for a given act relates to when the act was learned, and if the payoff for that act has subsequently changed (see 1.1.2 above), then the payoff level that the individual has recorded in its repertoire will be wrong.

1.2 Moves

1.2.1 Participants must specify a set of rules, henceforth a 'strategy', detailing when individuals should perform each of three possible moves. The options are:

1. INNOVATE (individual selects a new act at random from those outside its current repertoire, and learns that act and its payoff)
2. OBSERVE (individual selects another agent(s) at random, learn its (or their) act(s) and acquire an estimate of the relevant payoff or payoffs)
3. EXPLOIT (individual performs a specified act from its repertoire and reaps the payoff)

1.2.2 INNOVATE is equivalent to trial-and-error learning, and does not guarantee an improvement in available payoffs. INNOVATE selects a new act at random, from those acts not currently present in the individual's repertoire, and adds that act and its exact payoff to the behavioural repertoire of the individual. If an individual already has the 100 possible acts in its repertoire, it gains no new act from playing INNOVATE.

1.2.3 OBSERVE is equivalent to social learning. OBSERVE selects one or more other individuals at random, and observes the act(s) they performed in the last round, and an estimate of the payoff(s) they received. This knowledge is then added to the observing individual's repertoire. The number of other individuals sampled when playing OBSERVE is a parameter of the simulation, termed $n_{observe}$. In the pair-wise tournament phase $n_{observe}$=1; in the second phase we will run further conditions with $n_{observe}$>1. Note that individuals playing OBSERVE sample agents solely from among the subset that played EXPLOIT in the last round. If no individual played EXPLOIT in the last round then nothing is learned (see 3.5 below). Note also that it is possible for an individual to OBSERVE an act already in its repertoire, in which case only the payoff recorded for that act is updated.

1.2.4 Social learning is error prone with regard to both act and payoff. With a probability fixed to a single value between 0 and 0.5, the behavioural act returned by OBSERVE will not be that performed by the observed individual, but rather an act selected at random. Furthermore, the returned payoff estimate will be $\mu + \varepsilon$, where μ is the actual payoff of the observed individual and ε is a normally distributed random variable rounded to the nearest integer, with a mean of 0 and the standard deviation fixed to a single value between 0 and 10 (if $\varepsilon<0$ and $|\varepsilon|>\mu$ then the payoff estimate will be set to 0). These errors could represent the observation of migrants performing acts that are inappropriate in the current environment and/or mistakes in observational learning.

1.2.5 Individuals remember their own history of moves and payoffs, so strategies can access this information. Strategies can also, if desired, use this knowledge to update the payoffs stored in individual's repertoires over and above the updating described in 1.2.2-4.

1.2.6 EXPLOIT is the only move that results in a direct payoff to the acting individual (EXPLOIT here does not mean that another individual is taken advantage of, only that an individual is exploiting its knowledge). An individual can only EXPLOIT acts it has previously learned. When an individual chooses to EXPLOIT an act, the payoff it receives is used to update the payoff recorded in its repertoire (that is, we assume an individual can, by performing an act, update its knowledge, stored in its behavioural repertoire, of how profitable that act is).

1.3 Evolutionary dynamics: Lifespan, fitness and reproduction

1.3.1 Evolutionary change will occur through a death-birth process. Individuals die at random, with probability of 0.02 per simulation round giving an expected lifespan of 50 rounds, and are replaced by the offspring of individuals selected to reproduce with probability proportional to their mean lifetime payoffs. For individual *z*,

$$Pr(reproduction) = \frac{P_z}{\sum_i P_i}$$

where P_z is the mean lifetime payoff of individual *z* (that is, the sum of its payoffs from playing EXPLOIT divided by the number of rounds *z* has been alive) and the denominator is the summed mean lifetime payoff of the population in that round.

1.3.2 Offspring are behaviourally naïve: they have no behavioural acts in their repertoire and no knowledge of payoffs. Unless mutation occurs, offspring inherit the strategy of their parents. Mutation will occur with probability 1/50, and when it does, the offspring will have a strategy randomly selected from the others in that simulation. These mutations are how other strategies will first arise in a population initially containing only a single strategy. Mutation will not occur in the last quarter of each *melee* simulation (see 2.2 below).

2. Running the simulations

Details of how the simulations will run and how scores will be recorded in each evaluation stage are as follows:

2.1 Stage 1 (Pairwise contests)

Strategies will take part in round-robin contests against all other strategies. A contest involves each strategy invading a population of the other strategy. In a given simulation, a population of the dominant strategy will be introduced, and run for 100 rounds to establish behavioural repertoires. At this point, mutation will be introduced, providing the second strategy the opportunity to invade. Simulations will then run for up to a further 10,000 rounds. Each pairwise contest will be repeated 10 times with strategy A as the invader and 10 times with strategy B as the invader. The mean frequencies of each strategy in the last quarter of each run (i.e. the last 2,500 rounds in a 10,000 round run) will be averaged over the 20 repetitions. This average will then be recorded as the score of that strategy in that contest. Strategies will be assessed on their total score once every strategy has been tested against every other strategy.

2.2 Stage 2 (*melee*)

Simulations will start with an initial population consisting of individuals with a simple asocial learning strategy (INNOVATE once and then EXPLOIT on every subsequent move). Every time an individual reproduces, it has a 1/50 probability of mutating to a strategy chosen at random from the pool of 10 winners from stage 1. However, there will be no mutation in the last quarter of the simulation so that mutation does not unduly influence results when strategies have similar fitnesses. After 10,000 rounds, the mean frequency of each strategy in the last quarter of the simulation will be recorded as the score for that strategy. In addition to manipulating p_c, we will also vary the error rates associated with OBSERVE (the probability of learning an incorrect act will be drawn from the range [0 -0.5], and the standard deviation of ε, the error distribution of payoff observations, will be drawn from the range [1-10]), and the number of individuals observed for each OBSERVE move ($n_{observe}$ will be drawn from the range [1-6]). Simulations will be repeated 100 times for each of the conditions, and the strategy with the highest average score will be deemed the winner. The exact number of conditions we test will depend on computational constraints.

3. How to enter

3.1 Strategies will take the form of computer code functions that take the data specified below as arguments and return a decision on which move to play. An example strategy is given below. Strategies can be submitted as a Matlab function (using only those commands available in the base installation, excluding toolboxes), and/or ‘pseudocode’ (that is, linguistic instructions breaking down how decisions are made into a series of mathematical and logical operations that can each be directly translated into a single line of computer code[5]). If submitted as Matlab code, a pseudocode version should also be provided, to facilitate debugging. In all cases the code should be straightforward to translate between the formats. We provide in section 3.9 below an example strategy in both Matlab and pseudocode form and refer to that strategy by line number in the following descriptions.

3.2 The strategy function should return an integer number representing the individual’s move in this round (here termed `move`), and a 2-row array representing their behavioural repertoire (here termed `myRepertoire)`.

3.3 To play INNOVATE, `move` should be returned as -1; to play OBSERVE, it should be set to 0. Any positive integer greater than 0 will be interpreted as playing EXPLOIT, and the value of `move` will specify which behavioural act to perform (i.e. an integer value equal to one of the acts in the individual’s behavioural repertoire). This act must be present in the individual’s repertoire. If any individual tries to EXPLOIT an act not in its repertoire then it gets nothing for that round – no payoff and no addition to the behavioural repertoire. On the assumption that such attempts are mistakes in strategy algorithms, we will, for strategies submitted sufficiently before the deadline (see rules 8 and 11 below), attempt to contact the entrant(s) and invite them to revise their strategy, provided they do so before the entry deadline expires.

3.4 Each individual has access to its own behavioural repertoire. Strategies will be provided with this information in the form of a 2 by *n* array, where *n* is the number of acts in the repertoire, the first row of the array represents the acts themselves and the second row their payoffs. We assume that an individual can remember what it did over its lifetime, and how long it has been alive. Thus strategies will be provided with information on age, moves, acts exploited or learned, and the associated payoffs.

3.5 Strategies will receive the above knowledge in the form of three variables: `roundsAlive`, `myRepertoire` and `myHistory`. An individual that has survived 5 model rounds might receive the following data:

`roundsAlive = 5`	Number of previous rounds this individual has survived.

`myRepertoire = [19 2  64`
`                3  7  6 ]`
	The individual’s behavioural repertoire, containing three acts: 19, 2, and 64 (first row) with, according to the individual’s current knowledge, payoffs of 3,7 and 6

[5] For an example, see Mangel & Clark (2000) *Dynamic state variable models in ecology*. Oxford University Press, e.g. p55.

	respectively (second row).

`myHistory = [ 1  2  3  4  5`
`               0 -1  0  2  2`
`               2  19 64 2  2`
`               8  3  6  7  7]`
	Previous moves and their results – the first number in each column is the round to which that column pertains, the second is the move played (-1 for INNOVATE, 0 for OBSERVE, >0 for EXPLOIT), the third is the act learned (if OBSERVE or INNOVATE were played in that round) or exploited (if EXPLOIT was played), and the fourth is the payoff learned (OBSERVE or INNOVATE) or collected (EXPLOIT).

In this example case, $n_{observe}$ = 1. In its first model round, this individual played OBSERVE. As a result, it added act 2 to its repertoire and learned that this act returned a payoff of 8. In the second round it played INNOVATE, and added the act 19 with payoff 3 to its repertoire. In the third round it played OBSERVE and learned the act 64 with observed payoff 6. In rounds four and five, this individual played EXPLOIT, performed act 2, and received a payoff of 7. Note that its actual payoff received for act 2 was not exactly equal to the payoff learned for act 2 (when playing OBSERVE on the first round), because of the error in social learning (see 1.2.4).

Note also that in the case of new individuals, there will be no data – all the values shown above will be empty (i.e. of zero length) and if your strategy uses these inputs it should specify what to do in that case (and not crash!). The example strategy given below is robust to this as it specifically checks if `roundsAlive>1` (see section 3.9, line 2 of the MATLAB code).

3.6 Note that in above case, $n_{observe}$ = 1. If $n_{observe}$ = 3, for example, then the `myHistory` variable might look like this:

```
myHistory = [ 1  1  1  2  3  3  3  4  5
              0  0  0 -1  0  0  0  2  2
              2  86 10 19 64 2  0  2  2
              8  6  1  3  6  7  0  7  7]
```

Here, each OBSERVE move is represented by $n_{observe}$ (=3) columns in the `myHistory` variable, highlighted in bold. On the first OBSERVE move (round 1), the individual observed acts 2, 86 and 10 with payoffs of 8,6 and 1 respectively. On the second OBSERVE move (round 3) it observed two acts – 64, and 2 again. Note that there are two estimates here for the payoff associated with act 2 (8 in round 1, 7 in round 3) – these differences are due to the error in observing payoffs associated with OBSERVE (see 1.2.4 above). In the case that fewer than $n_{observe}$ individuals play EXPLOIT in the previous round, then some information returned by OBSERVE will be set to zero, as is shown for the underlined data from round 3 (zeros will also be returned if an individual that already has 100 acts in its repertoire plays INNOVATE). The behavioural repertoire for this individual in this case would contain acts 2, 86, 10, 19 and 64.

3.7 Strategies can choose to use their own rules to update their current knowledge of payoffs in the `myRepertoire` variable. However, strategies that do this must only change the second row of the matrix; the simulation engine will check that the repertoire of behaviours has not been altered. Strategies that do not update payoffs should return the `myRepertoire` array unchanged (this will happen automatically if the syntax used in the first line of the example strategy below is used). Payoff updates resulting from observing a behaviour already in the repertoire, or from exploiting a behaviour for which the payoff has changed, will be carried out automatically by the simulation program.

3.8 There are some rules that relate directly to the form of strategies; these are given below but also appear in the general tournament rules at the end of this document.

(1) There is no limit to the length of the function, but it cannot, on average, take more than 25 times as long as the example strategy, given in section 3.9, to reach a decision. If, on completion of the pair-wise tournament, this is found to be the case for your strategy, then it will not be eligible to win the tournament. However, if it proves to be an effective strategy, we may still discuss it in our reports of the tournament.

(2) Your strategy cannot access the disk or memory storage of the computer in any way beyond the information provided as input.

(3) Strategies playing EXPLOIT must specify which act to use from their repertoire. This act must be present in the individual's repertoire. If any strategy returns acts not in the repertoire then on the assumption that such attempts are mistakes in strategy algorithms, provided the strategy submitted sufficiently before the deadline we will attempt to contact the entrant(s) and invite them to revise their strategy, provided they do so before the entry deadline expires.

(4) Strategies modifying their own behavioural repertoires to update the stored payoffs can alter only those payoffs and not the list of acts stored. If any strategy attempts to do this, the same rules as in (3) will apply.

(5) We reserve the right to edit code for computational efficiency, but we will notify entrants if this occurs and they will be given the opportunity to check that the operation of their strategies has not been compromised.

(6) Strategies *must* be accompanied by both brief prose description of how they are intended to function and, if submitted as computer code, a 'pseudocode' version.

3.9 We provide below an example strategy to illustrate what is expected of entrants. This strategy is called ‘copy when payoff decreases’ (here given the function name ‘cwpd’). It starts by playing INNOVATE, and then EXPLOIT with the behaviour it learns. In subsequent rounds, it calculates the mean payoff the individual has received during its life, and if the last EXPLOIT returned less than that average, it plays OBSERVE and then EXPLOIT with the highest-payoff behaviour in its repertoire. This example illustrates how strategies must be prepared to handle the start of an individual’s life, when it has no acts in its repertoire, and how strategies can change as individuals survive over several rounds.

Matlab version (text in green are comments to aid interpretation):

```
function [move, myRepertoire] = cwpd(roundsAlive, myRepertoire, myHistory)
if roundsAlive>1; %if this isn’t my first or second round…
  myMeanPayoff = mean(myHistory(4,(myHistory(2,:)>0))); %mean payoff from EXPLOIT
  lastExploit = find((myHistory(2,:)>0),1,’last’); %find last EXPLOIT
  lastPayoff = myHistory(4,lastExploit); %get the payoff from last EXPLOIT
  lastMove = myHistory(2,find((myHistory(1,:)==roundsAlive-1),1,’first’)); %find last move
  if (lastMove==0) || (lastPayoff>=myMeanPayoff) %if lastMove was observe or lastPayoff at least as good as myMeanPayoff then EXPLOIT
    rankedR_Matrix = sortrows([myRepertoire’],-2); %rank acts by payoffs
    move = rankedR_Matrix(1,1); %perform the act with best payoff
  else %otherwise
    move = 0; %OBSERVE
  end
elseif roundsAlive>0; %if this is my second round…
    move = myRepertoire(1,1); %only have one behaviour from INNOVATE, so use that
else
    move = -1; %if this is my first round, then INNOVATE
end
```

Pseudocode version:

```
copy_when_payoffs_decrease:
1. If roundsAlive=0 then INNOVATE (move=-1)
2. If roundsAlive=1 then EXPLOIT with behaviour learned in first round (move = first value in myR)
3. If roundsAlive>1 then:
    4. Calculate my average myP value when myM>0, i.e. my average payoff when EXPLOITing, call it myMeanPayoff
    5. Find out when I last EXPLOITed, and store the payoff from that EXPLOIT as lastPayoff
    6. Find what my last move was, store as lastMove
    7. If lastPayoff<myMeanPayoff and lastMove<>0 (not OBSERVE) then OBSERVE (move=0)
       otherwise EXPLOIT by ranking acts by payoffs and choosing the highest ranking act (move=rankedActs(1))
```

4. Tournament Rules

1. Entry into the tournament must be accompanied by explicit acceptance of these rules.

2. The decisions of the committee shall in all cases be final and binding.

3. Anyone may enter the tournament, with the exception of current members of Kevin Laland’s research group, and members of the committee. Students of committee members are permitted to enter, but the committee will not be informed of entrant’s identities if they are asked to adjudicate specific issues.

4. Entrants may be single individuals, or collaborative groups. In the latter case, groups must select a corresponding entrant who will be the sole point of contact for the tournament organisers and the only person with whom the organisers will discuss that entry, and also provide a list of the group members. Entrants may submit only one strategy, and individuals may only participate in one group entry.

5. Only entries received by 1700 GMT on the closing date, 30th June 2008, will be accepted, and no further modification of entries is permitted after this date. Entrants are *strongly* advised to submit their strategies well before this time so that the organisers can check the code and inform entrants of any problems before the final closing date (see 8 and 11).

6. All entrants agree to the *content* of their submission being made public as part of the communication of this research exercise, although entrants can choose not to have their *name* associated with their entry.

7. There is no limit to the length of the function, but it cannot, on average, take more than 25 times as long as the example strategy, given in section 3.9, to reach a decision. If, on completion of the pair-wise tournament, this is found to be the case for your strategy, then it will not be eligible to win the tournament. However, if it proves to be an effective strategy, we may still discuss it in our reports of the tournament.

8. Your strategy cannot access the disk or memory storage of the computer in any way beyond the information provided as input. The organizers reserve the right to disqualify strategies that are deemed not in the spirit of the contest.

9. Strategies playing EXPLOIT must specify which act to use from their repertoire. This act must be present in the individual's repertoire. If any strategy returns acts not in the repertoire then on the assumption that such attempts are mistakes in strategy algorithms, provided the strategy submitted sufficiently before the deadline (see 11 below) we will attempt to contact the entrant(s) and invite them to revise their strategy, provided they do so before the entry deadline expires.

10. Strategies modifying their own behavioural repertoires to update the stored payoffs can alter only those payoffs and not the list of acts stored. If any strategy attempts to do this, the same rules as in (3) will apply.

11. We reserve the right to edit code for computational efficiency, but we will notify entrants if this occurs and they will be given the opportunity to check that the operation of their strategies has not been compromised.

12. Strategies *must* be accompanied by both brief prose description of how they are intended to function and, if submitted as computer code, a 'pseudocode' version.

13. Entrants must be prepared to enter into a reasonable dialogue with the organisers to remove ambiguities from the entered strategy for the purposes of coding the simulations and improving computation efficiency. We will endeavour to inform entrants if their strategies do not function correctly. If a strategy is deemed inadmissible prior to the closing date, entrants will be informed forthwith and given the opportunity to revise their submission. Strategies deemed inadmissible after the closing date will be disqualified. We guarantee to test any strategies submitting up to one month before the closing date; we can give no such guarantee for strategies submitted after this time, although we will endeavour to do so.

14. If a strategy is submitted that, in the opinion of the organisers, is so similar to one already submitted as to be reasonably considered identical, then the first submission will take precedence and the submitter of the identical strategy will be informed that their entry is ineligible. The submitter will however be eligible to revise and resubmit their entry, provided that they do so prior to the closing date.

15. Any strategy that, in the opinion of the organisers, has been designed so as in any way to recognise and specifically help other entered strategies at their own expense will be disqualified and the authors of the strategy will be given no further opportunity to enter a modified strategy. This rule is essential to preserve the evolutionary validity of the tournament.

16. In the event of a tie, the tied strategies will be submitted to further tests under varied simulation conditions as deemed appropriate by the organising committee. If the committee judges that the tied strategies do indeed have equal merit, then they may decide at their discretion to share the prize between the tied entrants.

17. In the event that the number of submitted strategies renders a complete set of pairwise contests computationally unfeasible, we reserve the right to use a different system to select which strategies move forward to the *melee* stage, for example by splitting the strategies into randomly assigned groups from which winners will be selected to go forward to the *melee* stage.

18. The winning entrant will receive a cash prize of 10,000 Euros to be presented at a conference organised around the tournament at the University of St Andrews in 2009. The winning entrant will also be invited to co-author a paper reporting on the contest, although the organizers reserves the right to produce the paper without the entrant’s participation and to judge authorship merits at their discretion.

References for Rules of Entry

Axelrod, R. 1984. The Evolution of Cooperation. New York: Basic Books.

Barnard, C. J. & Sibly, R. M. 1981. Producers and scroungers – a general-model and its application to captive flocks of house sparrows. Animal Behaviour, 29, 543–550.

Boyd, R. & Richerson, P. J. 1985. Culture and the Evolutionary Process. Chicago: Chicago University Press.

Boyd, R. & Richerson, P. J. 1995. Why does culture increase human adaptability? Ethology and Sociobiology, 16, 123-143.

Enquist, M., Eriksson, K. & Ghirlanda, S. 2007. Critical social learning: a solution to Rogers' paradox of non-adaptive culture. American Anthropologist, in press.

Feldman, M. W., Aoki, K. & Kumm, J. 1996. Individual Versus Social Learning: Evolutionary Analysis in a Fluctuating Environment. Anthropological Science, 104, 209-231.

Galef Jr., B. G. 1995. Why behaviour patterns that animals learn socially are locally adaptive. Animal Behaviour, 49, 1325-1334.

Giraldeau, L.-A., Valone, T. J. & Templeton, J. J. 2003. Potential disadvantages of using socially acquired information. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 357, 1559-1566.

Henrich, J. & McElreath, R. 2003. The evolution of cultural evolution. Evolutionary Anthropology, 12, 123-135.

Laland, K. N. 2004. Social learning strategies. Learning and Behaviour, 32, 4-14.

Rogers, A. 1988. Does biology constrain culture? American Anthropologist, 90, 819-813.

Schlag, K. H. 1998. Why imitate, and if so, how? Journal of Economic Theory, 78, 130-156.

Appendix B: Details of tournament entries competing in first stage.

Rank	Strategy Name	First stage score - multiple conditions	First stage score - single condition	Department, Country
1	copyWhenYoungThenLearnWhenPayoffsDrop	0.75	0.88	Department of Primatology, Max Planck Institute for Evolutionary Anthropology, Germany
2	dynamicAspirationLevel	0.73	0.87	Philosophie, Germany
3	prospero	0.72	0.86	Department of Physics and Astronomy, Canada.
4	discountmachine	0.69	0.89	Department of Mathematics and Statistics, Canada
5	intergeneration	0.68	0.85	Faculty of Mathematics and Physics, Czech Republic
6	valueVariance	0.64	0.76	None given, USA
7	wePreyClan	0.63	0.79	Department of Ecology & Evolutionary Biology, USA
8	rummer	0.59	0.80	Department of Sociology, Netherlands
9	whenTheGoingGetsToughGetScrounging	0.54	0.84	Westminster School Sixth Form, UK
10	livingdog	0.51	0.77	Dipartimento di Studi Sociali, Italy
11	stabilityObserver	0.50	0.72	None given, Germany
12	w00t	0.50	0.76	University of California, USA
13	senescence	0.47	0.79	Biology, Denmark
14	progressivepeakseeker	0.46	0.77	Mathematics, USA
15	evchooser	0.46	0.74	Computer Science, USA
16	keepUp	0.43	0.77	None given, USA
17	improvedCwpd	0.38	0.75	None given, Sweden
18	indecisiveJDK	0.38	0.76	None given, USA
19	halfmax	0.34	0.73	Physics, Netherlands
20	observe3ThenExploit	0.34	0.78	Computer Science, USA
21	learnAtTheBeginningThenExploit	0.34	0.74	Stockholm Resilience Centre, Stockholm University, Sweden
22	weightedContextAware	0.34	0.73	None given, USA
23	startExploitRecover	0.29	0.73	Centre for the Study of Cultural Evolution, Sweden
24	firstLookThenExploit	0.28	0.74	School of Human Evolution and Social Change, USA
25	oneThirdSocial		0.70	None given, USA
26	whoDoISee		0.69	None given, Netherlands
27	whatYouSeeIsWhatYouDo		0.68	Ecole des Hautes Etudes Commerciales, Switzerland
28	aHandfulOfSkills		0.67	College of Engineering, USA
29	lookahead		0.67	Computer Science, USA
30	instancebased		0.66	Computer Science, USA
31	gatherDataAndHillClimb		0.65	None given, USA
32	breakthroughInnovation		0.65	None given, Canada
33	learnFromOthers		0.65	None given, Germany
34	spyNWork		0.64	None given, Germany
35	waitForSomethingBetter		0.64	None given, USA
36	followTheMeans		0.64	Center for Adaptive Behavior and Cognition, Max Planck Institute for Human Development, Germany
37	sabbath		0.62	Department of Evolutionary Biology, Czech Republic
38	copyMoreWhenYounger		0.62	Computer Science and Artificial Intelligence Laboratory, USA
39	divideAndConquer		0.62	Department of Anthropology, USA
40	rebelWithoutACause		0.60	Department of Anthropology, USA
41	adaptiveControl		0.60	Meiji Institute for Advanced Study of Mathematical Sciences (MIMS), Japan
42	cUDOS		0.59	Département des Sciences Biologiques, Canada
43	hydra		0.59	None given, USA
44	copyAndSwitch		0.57	L'Institut de recherche pour le développement, France
45	carefullyRecalcDude		0.57	None given, USA
46	stCoop		0.56	None given, France
47	itTakesAVillage		0.56	None given, USA
48	lateAdopter		0.56	Department of Developmental and Comparative Psychology, Max Planck Institute for Evolutionary Anthropology, Germany
49	copyWhenConditionsAreStable		0.55	Département des Sciences Biologiques, Canada

Rank	Strategy Name	First stage score - multiple conditions	First stage score - single condition	Department, Country
50	goldberg		0.55	Department of Biological and Environmental Sciences, Finland
51	bestMeanSelect		0.55	None given, Finland
52	roman5		0.55	None given, Canada
53	weightedObserver		0.55	Mathematics, Sweden
54	herculesAtTheCrossroads		0.54	Department of Developmental and Comparative Psychology, Max Planck Institute for Evolutionary Anthropology, Germany
55	indexBasedLearningDecision		0.54	Mathematics Mathematics, USA
56	learnFirst		0.53	None given, USA
57	marmoset		0.52	Département Ecologie, Physiologie & Ethologie, France
58	infantJuvenileMature		0.52	Biological Sciences, USA
59	julieDecstep25		0.51	Département des Sciences Biologiques, Canada
60	lowVarianceObserver		0.51	Department of Social and Developmental Psychology, UK
61	estimatePcThenChooseStrategy		0.50	Mathematics, Sweden
62	copyWhenPayoffsDecreasePlus		0.49	Department of Innovation and Environmental Sciences, Netherlands
63	staticLearningReduction		0.48	None given, USA
64	wiseOrObserve		0.47	None given, UK
65	greatExpectations		0.47	Institute of Cognitive and Evolutionary Anthropology, UK
66	kISAgent		0.47	Faculty of Mathematics and Natural Sciences, Netherlands
67	updateWhatYouHave		0.46	Département des Sciences Biologiques, Canada
68	learnUntilProfitable		0.45	Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Belgium.
69	observeEarlyInnovateLittle		0.41	None given, USA
70	infoScrounger		0.40	Département Ecologie, Physiologie & Ethologie, France
71	weightedSocialLuceChoice		0.37	Psychology, USA
72	optimalStopping		0.36	Centre for Behavioral Biology, UK
73	tangle7		0.35	Behavioral Biology, Netherlands.
74	twiceAsGood		0.33	Biology, USA
75	goodacts		0.32	Computer Science, USA
76	magpie		0.31	Centre for Behavior & Evolution, UK
77	criticalSocialLearner		0.31	Psychology, Italy
78	knowledgeWeighters		0.31	None given, Portugal
79	adaptolution		0.30	Wells Cathedral School, UK
80	innovateBeforeObserveInRadicalEnvironmentalChange		0.30	Institut für Sozialwissenschaften – Soziologie, Germany
81	monkeyStudent		0.29	Psychology, Canada
82	smartObserve		0.29	Computer Science, USA
83	mainlyObservers		0.28	None given, USA
84	ratchet		0.28	Psychology, UK
85	genderedStrategy		0.27	None given, USA
86	innovateAndObserve		0.26	None given, Netherlands
87	constantProbability		0.22	School of Electronics and Computer Science, UK
88	piRounds		0.22	Zoology, Sweden
89	kiss		0.20	Economics, USA
90	econoSearch		0.19	ETS Ingenieros Industriales, Spain
91	copyIfBetter		0.16	Anthropology, UK
92	weightedExploitation		0.15	None given, USA
93	noImitator		0.15	Management Department, USA
94	smashNgrab		0.14	Wells Cathedral School, UK
95	higherLearning		0.13	Biology, Canada
96	prospector		0.11	None given, USA
97	aynRandGambit		0.11	None given, Canada
98	unobservant		0.10	Dept of Medical Education, USA
99	randomness		0.09	Howe School of Technology Management, USA
100	copyWhenPayoffsDecrease		0.07	None given, Germany
101	balancedCopyWhenPayoffsDecrease		0.06	None given, USA
102	exploitOneInnovation		0.05	Behavioral Biology, Netherlands
103	stateDefined		0.04	None given, USA
104	observeNoThanks		0.02	None given, France

Appendix C: Details of tournament entries in second stage.

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
1	*discountmachine*	0.35	Department of Mathematics and Statistics, Queen's University, Canada	Our creature does three major things: First it estimates/calculates, what we believe to be all the pertinent parameters of the simulation as well as a few other quantities that we believe to be useful. These are P_c, the mean of the payoff distribution, the mean of of the observed values, the correlation between observe and exploit values of the same action, N_observe, and where applicable the number of data points used to make these estimates. Second it uses some of these parameters to estimate the expected payoff for performing each action in its repertoire. Once it has a best exploit chosen from its repertoire it compares the value of Exploiting to the value of Observing using a geometric discounting scheme based on our estimate of P_c and the given probability of death. Lastly a machine learned function, takes into account N_observe and the estimates on the reliability of observing and P_c to adjust the value of Observing accordingly. Our creature then chooses whichever action has the higher perceived value, Observing or Exploiting. As a side note our creature only Innovates when it has an empty repertoire and observe doesn't work, which typically is only on the first turn of a simulation.
2	*intergeneration*	0.23	Faculty of Mathematics and Physics, Charles Unviersity, Czech Republic	My main idea is (although it seems not be as good as I expected) that an important information for the young is, how much is the "good" payoff (with how much I can be happy). If I have so much or more, I would just EXPLOIT until it changes, otherwise I would 8 times exploit and once observe. The important trial is that the old could "say" something to the young, by "signaling" something to the young. The signal consits of doing an act whose number is divisible by 8. If the fraction is 1,2,3,4 this means that "payoff 8 is very good", 5,6,7,8 means "payoff 20 is very good" and 9,10,11,12 means payoff 40 is very good. If the old does not have this in his repertoire, he innovates. If he has more than one of these 4 possible "symbol" acts, he uses that with highest payoff -- because it is a higher chance that this will diffuse. Even

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				the "opponent" can help spread out this signal, wihout knowing that.
3	*prospero*	0.09	Department of Physics and Astronomy, McMaster University, Canada.	1. Prospero estimates the mean payoff by taking the mean of all the elements in the 4th row of its history. All elements in this row are included in the mean, no matter whether they are actual payoffs received from exploiting or payoffs learned by innovating or observing. 2. Prospero determines the best act in its current repertoire by finding the highest value in the second row of MyRepertoire. When Prospero exploits, it always exploits the best act in its repertoire. In the case where more than one act in the repertoire has the same best payoff in the second row of MyRepertoire, Prospero exploits one of the acts at random from among those acts whose payoff is equal to the best payoff. 3. Prospero compares its last payoff with the mean payoff. If the last payoff is higher than the mean, then Prospero is satisfied, and it continues to exploit its best act. Furthermore, Prospero never observes or innovates twice in a row, so if the last move was observe or innovate, Prospero always exploits. 4. If the last move was exploit and the last payoff was less than or equal to the mean payoff, Prospero is not satisfied. It then chooses its move randomly with probabilities controlled by two parameters a and b. Observe with probability ab, Innovate with probability a(1-b), Exploit current best act with probability (1-a). I will refer to observe plus innovate together as learning. The parameter a controls the ratio of learn to exploit, and the parameter b controls the ratio of observe to innovate, given that a learning move is played. 5. Prospero estimates the probability of change of the environment, pc, from its history. It chooses to learn with a low probability when pc is low because it is very likely already exploiting a high-payoff act, so it does not want to waste a move on learning. It chooses to learn with a high probability when pc is intermediate because it is important to learn in order to keep up with changes in the environment. It chooses to learn with a low probability when pc is very high because learning has little value if the payoff changes frequently. Prospero counts nsame , the number of times that it exploited the same act as the last round and obtained the same payoff as the last round, and ndiff , the number of times that it exploited the same act as the last round and obtained a different payoff to the last round. 6. Prospero estimates the diversity of strategies being used in the population from its previous observations. It chooses to observe more frequently (i.e. larger value of b) when its estimate of diversity

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				is high, and to innovate more frequently when its estimate of diversity is low. The rationale is that when the diversity is high it is best to observe, because the acts being performed by others will be much better than random acts, whereas it is not so useful to observe when the diversity is low, because most likely you will observe something that is already in the repertoire. Let nrep = number of acts in the current repertoire, n0 = number of observations made before (= number of 0’s in row 2 of history), and n1 = number of innovations made before (= number of -1’s in row 2 of history). The number of different strategies that have been observed is nrep – n1. This is less than n0 when the diversity is low, because the same act has been observed more than once. Prospero uses the following rule to set the parameter b: nrep+n0-n1 / nrep+n0. In the example history above, nrep = 5, n0 = 5, and n1 = 3. Therefore b = (1+5-3)/(1+5) = 0.5. In the example history, the number of acts observed on one turn is nobserve = 1, so n0 = number of turns on which observe was played. In the case where nobserve > 1, n0 = nobserve × number of turns on which observe was played. However, the same formula for b is applicable in either case. 7. Special rules apply at the beginning of Prospero’s life when it has no strategies in its repertoire. On the first turn it always observes, on the grounds that an observed act will have a much higher payoff than a randomly innovated act. If no act is observed on the first round (because no individuals exploited), Prospero innovates on the second round. This case will only arise at the beginning of the simulation when all individuals are started at the same time. Usually a new individual will be born into a population of adults, so there will always be something to observe on the first round.
4	*wePreyClan*	0.07	Department of Ecology & Evolutionary Biology, University of California, USA	In the first round (roundsAlive = 0), the strategy observes. In round two, if no one was observed EXPLOITING in round one, the strategy INNOVATES. If individuals were observed, the strategy chooses the best payoff from myRepertoire, and EXPLOITS it. After the second round, the strategy becomes a hybrid of 3 different tactics. If n=1 (OBSERVE one exploiter at a time), the strategy calculates from myHistory the probability that in subsequent rounds the payoff for a given act (either OBSERVED, EXPLOITED, or INNOVATED) has changed. If this probability is >0.6 (and there at least 5 data points to calculate the probability), the individual chooses the act with the highest payoff from myRepertoire and EXPLOITS. If the probability is <= 0.6, then the strategy looks through myHistory to find the greatest recorded payoff (myHistory(4,:)). If the individual has EXPLOITED for the last two rounds (t-1, t-2) and the payoff from the last round (t-1) was less than (0.45 * the maximum payoff value in myHistory), then the individual will OBSERVE, otherwise it will re-sort myRepertoire and then play the top act from there.

2^{nd} Stage Rank	Strategy Name	2^{nd} Stage Score	Address	Strategy description (as submitted by the strategy authors)
				If n>1, then the strategy calculates from myHistory the mean and median payoff for all acts (either OBSERVED, EXPLOITED, or INNOVATED). Whichever of the two values is greater is set to be MHigh, and the lesser is MLow. The strategy also calculates a correction factor (CF = 1 – 0.2(1 – n/6)). Every 10th roundsAlive, if the strategy EXPLOITED in the previous round and the payoff was < MHigh*CF, the strategy switches to OBSERVE. Otherwise, it EXPLOITS the act from myRepertoire with the highest payoff. For the other 9 rounds the strategy will shift from EXPLOIT to OBSERVE if previous payoff was < MLow*CF. Thus, every 10th round the strategy becomes more likely to OBSERVE.
5	*valueVariance*	0.06	No affiliation given, Texas, USA	Rounds 0 & 1 are obviously Innovate then Observe. The next paragraph checks to make sure that I have at least two values in my repertoire so that I can start calculating the best move. If it doesn't have two values then it either innovates or observes until it get two values. Once we get past that section we do some math to try to determine the distribution of the payoff values. ExploitChance gets set to a high value if the highest payoff amount in the repertoire is significantly higher than the mean value of payoffs. In the end, the strategy assigns a percentage chance from 1-100 for each of the possible actions. If ExploitChance is > 100 then it's definitely going to exploit that highly valued act. When ExploitChance is <100 the left over percentage (aka NonExploitChance) chance is split up between Observe and Innovate. A variable named ObservationMultiplyer is based on the number of observations made when an OBSERVE is performed (constrained to 3). The ObserveChance is calculated and then a function adds a little more based on the ObservationMultiplyer. Now that we know the percent likelihood of Exploiting and Observing, everything else is innovating. So, the last section finds a random number and determines if it is within ExploitChance, greater than ExploitChance but less than ExploitChance plus ObserveChance, or greater.
6	*dynamic-AspirationLevel*	0.06	Universität Bayreuth, Philosophie II, Germany	DynamicAspirationLevel determines an aspiration level that is slightly higher than the mean payoff of all previous EXPLOIT moves. The difference between the maximun payoff in the repertoire and the aspiration level determines the probability with which an OBSERVE move is chosen vs. an EXPLOIT move. Furthermore, "DynamicAspirationLevel" uses a 'decaying repertoire' where old entries and potentially invalid entries are values less than new ones. The strategy in detail: The strategy "DynamicAspirationLevel" consists of two phases, the first of which lasts until the fourth round of the life time of an individual using this strategy. The second phase last during the remainder of the life time of

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				the individual. 1. Phase One: - Play OBSERVE in the first round alive (when 'roundsAlive = 0') - In the second round alive: If the repertoire is still empty (i.e. nothing could be oserved in the first round) play INNOVATE, otherwise chose the best EXPLOIT move from the repertoire. - In the third round alive, play OBSERVE - In the fourth round, chose the best move from the repertoire 2. Phase Two - determine the mean payoff 'm' from all EXPLOIT moves in the history. - determine the maximum payoff 'M' from all EXPLOIT moves in the history. - calculate an aspiration level 'a' according to the magic formula: a := m + (M-m) / 5 - determine the highest payoff 'ace' in the repertoire. - if the highest payoff in the repertoire is greater or equal the aspiration level ('ace >= a') then set the observataion probability 'p' to 0.01. Otherwise, if the aspiration level 'a' is greater zero, let the observation probability 'p' be the maximum of 0.01 and the difference between the aspiration level and the highest payoff in the repertoire, divided by the aspiration level '(a-ace)/a'. Otherwise, if the aspiration level is zero, let 'p' be one. - generate a random number 'r' between 0 and 1. If the random number is less or equal the observation probability 'p' chose an OBSERVATION move with a chance of 99% and an INNOVATE move with a chance of 1%. Always choose INNOVATE if the aspiration level 'a' was not greater zero! Otherwise chose the best EXPLOIT move from the repertoire. - Decay the repertoire by subtracting 1 from each payoff value registered in the repertoire.
7	*copyWhenYoungThenLearnWhenPayoffsDrop*	0.06	Max Planck Institute for Evolutionary Anthropology, Department of Primatology, Germany	At the beginning of their life individuals always observe two times. In this way the individual gets good knowledge about what is worth to exploit. The second observe accounts for the problem that observing is error prone and that the environment maybe just changed. Afterwards individuals always exploit the trait with the highest known pay-off as long as the highest known pay-off is larger then 80 % of the average of all pay-offs that were ever experienced (exploited, observed and innovated). In this way the individuals tend to exploit only traits with high payoffs without getting too choosy. In case the highest known payoff drops too strongly individuals first always innovate, afterwards they always observe once

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				and then randomly decide to either innovate or observe. To always innovate first is done because the environment just changed and in this case observing might provide outdated information. In the next time step this is not the case anymore, therefore the individuals should observe.
8	*livingdog*	0.04	Dipartimento di Studi Sociali, Università di Brescia, Italy	Living Dog is a probabilistic strategy based on two parameters: p and q, where p is the probability of playing EXPLOIT (vs. INNOVATE or OBSERVE) and q is the probability of playing OBSERVE (vs. INNOVATE). The parameter values are defined depending on the agent repertoire and history. More specifically, p is computed as a fucntion of the differences among the sorted payoffs included in the agent repertoire. The underlying assumption is that a jump between two payoffs much larger than the median one implies that the agent already knows an act associated with a high payoff. The value of q depends instead on the relative value of the average earnings when the agent exploited an act learned using INNOVATE and of the average earnings when the agent exploited an act learned using OBSERVE. The function uses a two step decision making process: 1) a random extraction based on p determines whether the agent will EXPLOIT its higher act or try to increment its repertoire; 2) if it does not EXPLOIT, a random extraction based on q determines whether it will OBSERVE or INNOVATE (actually, in order to increase the function speed, the value of q is computed only when the 1st step does not return an EXPLOIT decision).
9	*rummer*	0.02	Department of Sociology / ICS, Utrecht University, Netherlands	The main idea behind RUMmer is that successful behavior in an uncertain environment, and in the presence of other "competitors", must be adaptive. A strategy must be able to adapt to different environmental conditions, for example, how hard it is to attain relatively high rewards, how fast is the environment changing. The relative returns on innovating, observing others, and exploiting what you already know depend also on the strategies of others. They might be asocial innovators, but perhaps conformist social learners. Therefore, in addition to the properties of the environment, we conjecture that a successful strategy should also be able to adapt to the types of strategies that its competitors use. RUMmer tries to bring these considerations together by using propensities for possible moves: INNOVATE, OBSERVE, or EXPLOIT. Propensities are calculated from agent's experience of whether

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				and to what extent each of these moves was beneficial. An agent should innovate more often, the higher the benefits are of the acts he learned by innovating. Analogously, an agent should observe more often if benefits of observed acts are high. Another important aspect of RUMmer is that it is probabilistic. Calculated propensities determine the probabilities of playing each of the three moves in every round. We believe that randomizing the move in every round is an advantageous way to behave in a probabilistic (uncertain) environment. Deterministic strategies may run into trouble as in every round some aspect of the environment is likely to change. Thus, beneficial options calculated from past rounds may not be beneficial later on. Finally, RUMmer implements an idea that it can only exploit its knowledge during exploitation rounds. Innovations and observations have an "opportunity cost" equal to the payoff that could have been earned in that round. Thus, the benefits of innovation and observation are reduced with the actual likelihood of exploitation. The basic mechanism is the following. RUMmer INNOVATEs in the first round, EXPLOITs in the second, and OBSERVEs in the third. From the fourth round on, the following quantities are calculated in every round: 1. Highest known payoff based on myRepertoire (hexp) 2. Mean payoff of innovated actions based on myHistory (minn) 3. Mean payoff of observed actions based on myHistory (mobs) 4. Proportion of rounds in which played EXPLOIT (pexp) Additionally, let denom be equal to: exp(pexp*minn) + exp((pexp*mobs) + exp(hexp) Then choose one of the three possible moves with the following probabilities: P(INNOVATE) = exp(pexp*minn) / denom P(OBSERVE) = exp(pexp*mobs) / denom P(EXPLOIT) = exp(hexp) / denom Playing EXPLOIT is choosing the action with the highest utility in myRepertoire.
10	*whenTheGoing-GetsToughGetScrounging*	0.02	Westminster School Sixth Form, UK	In general, after observing twice to learn (at least) 2 acts, the strategy exploits 4 times in order to get some early payoff and decide if Pc is low or high. If it sees an act's payoff change in these 4 moves it decides Pc is high, and observes on move 7 to indicate this decision to itself in future. Otherwise it

2nd Stage Rank	Strategy Name	2nd Stage Score	Address	Strategy description (as submitted by the strategy authors)
				exploits. From move 8 on it looks at move 7 to remind itself whether it decided Pc was low or high. If it decided Pc was high it exploits the highest payoff act unless this is less than the mean of payoffs it has seen, in which case it exploits the act it hasn't used for longest. If it decided Pc was low it exploits highest unless the last move was an exploit with a payoff lower than its mean lifetime payoff. In this case it observes. It will also always observe on move 8. If its last move was an observe it checks to see if the act that was observed was the same as the act exploited on the round before the observe. If this is the case it innovates, since it is clear that more acts need to get into circulation. There are two exceptions to this mode of behavior. Firstly, some small changes are made to the opening made if the Observe on move 1 fails due to nobody exploiting in the previous turn. Secondly, if Nobserve is found to be greater than 2, move 7 becomes an exploit even if it decides Pc is high. The strategy will then continue to act in the mode which it does normally if it had in fact decided that Pc were low. "Nobserve greater than 2", both here and in the pseudocode, means strictly greater than 2, not greater than or equal to 2.