Supporting Information for: Landowner preferences and conservation prioritization: response to Nielsen et al.

S1 – Materials and methods

<u>Simulation of the data:</u>

The simulation of the data was accomplished by a short python script. We attempted to create simulated data which may be considered to be comparable to the real data from Nielsen et al. (2017), however our simulated data does not capture all of the details of the real data. We considered three pieces of information to be essential to re-create the simulated data: the available area for conservation in each cell, the current presence probability for the set of species in each cell, and the cost for conserving each cell.

The simulations were conducted using very simple probability models (Appendix 1). To reflect the basic properties of the data we used the basic distribution data available in the article (Nielsen et al. 2017). In their supplementary data, information regarding the basic statistics of each cell was provided in Table Appendix S1.2. The area of the uninformed forest available for conservation was reflected by a re-creation of the basic quartile distribution. For each cell, a random number was generated to first allocate the cell into a specific quantile, a second random number then set the area within the range of the quantile. The last quantile (75-100%) due to the apparent kurtosis was treated in a slightly different fashion, 21% of the values were required to be within the range of (1,500 and 2,500 ha) and 4% of the values were within the range of (2,500 and 6,870 ha). The informed area available for conservation was simply a percentage reduction of the areas. For cells with larger forest areas, the percentage removed was less than for cells with small forested areas. The total reduction from the uninformed case was approximately 50%.

The current presence probability for each species depended upon the current forest area available (uninformed) for conservation. The total number of species available at each cell was derived from Figure Appendix S2 from Nielsen et al. 2017). Cells were allowed to allocate species probabilities for a range of species depending on the total area (table S1). The actual probabilities for the species were evaluated using a common probability model. Twenty percent of the species had a survival probability of greater than 95%, 30% of the species had a survival probability of between 80 and 95%, 30% of the species had a survival probability of between 50 and 80% and 20% of the species had a survival probability of between 10 and 50%.

The cost of conservation was evaluated using a very simple model. A random number for each cell was selected between 1700 and 3000, to reflect the range of costs from what was obtained by Nielsen et al. (2017). As the simulated data was generated by a simple python code, we have included the code as an attachment to this supplementary material.

<u>Optimization model:</u>

The optimization model used in this analysis is the supplementary material S4 of in Nielsen et al. (2017). As the survival curves were only provided as a figure, we estimated both the approximate linear and sigmoidal survival curves through a visual approach. The CPLEX model is included as in appendix 2. All of the variants to the model are included through the introduction of 2 parameters, one parameter (UNINFORMED) controls for using the uninformed or informed case, while the parameter (SIGMOID) controls for the sigmoid or approximate linear survival curves. To highlight that models generate similar results to Nielsen et al. (2017), Figure S1 highlights the efficiency improvements by incorporating the conservation preferences of the forest owners.

Table S1. The species occurrence treated as a function of the initial forested area. Species occurrence decreases with decreasing forested area.

Area	Range of species allocated
>2,000	68-116
1,500 - 2,000	49-67
1,000 - 1,500	33-48
500 - 1,000	17-32
< 500	0-16

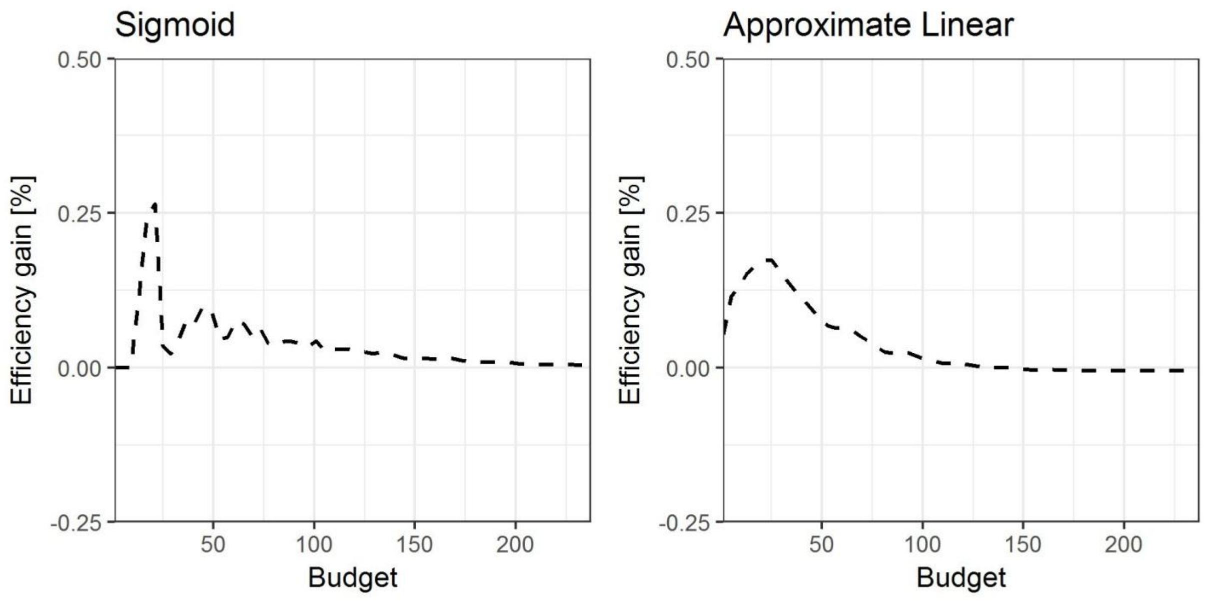

Figure S1. The expected efficiency gain for the species probability functions when using the informed case over the uninformed case. Positive values indicate improvements due to incorporating landowner preferences.

References:

Nielsen ASE, Strange N, Bruun HH, Jacobsen JB. 2017. Effects of preference heterogeneity among landowners on spatial conservation prioritization. Conservation Biology. 31(3): 675-685.

Appendix 1. Python code for creating the simulated data for use as a data file for CPLEX.

```
import random

def areas(cells):
    informed = []
    uninformed = []
    # A probabilistic model to generate a range of forested areas for each cell.
    # The proportion of areas represent the data provided in NSBJ's
    # supplemental material.
    for br in range(0,cells):
        rr = rr = random.random()
        if rr > .96:
            r = float(random.randrange(2500,6870))/10000
        elif rr >0.75:
            r = float(random.randrange(1500,2500))/10000
        elif rr >0.5:
            r =float(random.randrange(786,1500))/10000
        elif rr >0.25:
            r = float(random.randrange(275,786))/10000
        else:
            r = float(random.randrange(1,275))/10000

        uninformed.append(r)
        # A simplistic model to identify how much of the cell will be conserved,
        # for the informed case.

        # informed area is at least +30% to +70% of informed)
        if r> 0.1500:
            informed.append(r*(1-float(random.randrange(200,500))/1000))
        elif r> 0.0786:
            informed.append(r*(1-float(random.randrange(300,600))/1000))
        elif r> 0.0275:
            informed.append(r*(1-float(random.randrange(400,700))/1000))
        else:
            informed.append(r*(1-float(random.randrange(500,800))/1000))

    return informed , uninformed

def probabilities_sp(species,area):
    one_row = []
    #Species occurrence probability dependent on the current forested area.
    #A simplified approach to that used by NSBJ.
    #Species occurance increases with increased forested area
    if area > 2000:
        rr_range = random.randrange(68,116)
    elif area > 1500:
        rr_range = random.randrange(49,67)
    elif area > 1000:
        rr_range = random.randrange(33,48)
    elif area > 500:
        rr_range = random.randrange(17,32)
    else:
        rr_range = random.randrange(0,16)

    #
The specific probability for each species generated by a probabalistic model.
```

```python
    # A if statement is used to evaluate if the species will be given
    # a probability or not.

    for br in range(0,species):
        if random.random() > float(rr_range)/170:
            rr = random.random()
        else:
            rr = 0
        if rr >0.8:
            one_row.append(float(random.randrange(950,1000))/1000)
        elif rr >0.50:
            one_row.append(float(random.randrange(800,950))/1000)
        elif rr >0.2:
            one_row.append(float(random.randrange(500,800))/1000)
        else:
            one_row.append(float(random.randrange(100,500))/1000)
    return one_row

def costs(cells):
    one_row = []
    for br in range(0,cells):
        # Range of opportunity costs per ha of a cell.
        # Ranges between 1,700 and 3,000 DKK per ha.
        rc = random.randrange(1700,3000)
        one_row.append(rc)
    return one_row

# A main function with incorporates the data into a cplex dat file.
def main(cells, species, file_name):

    informed, uninformed = areas(cells)
    # Creating data file for CPLEX model.
    g = open(file_name, 'w')
    g.write(str("stands = "+str(cells)+";\n"))
    g.write(str("species = "+str(species)+";\n"))
    g.write(str("Bk = "+str([0.000001]+[x*0.0025 for x in range(1,401)])+";\n"))
    g.write(str("c = "+str(costs(cells))+";\n"))
    g.write(str("area_in="+str(informed)+";\n"))
    g.write(str("area_un="+str(uninformed)+";\n\n"))
    g.write(str("pji_un = ["))

    for br in range(0,cells):
        g.write(str(probabilities_sp(species,uninformed[br])))
        g.write(str("\n"))
    g.write(str("];"))
    g.write(str("pji_in = ["))

    #To evalaute the species occurence probability of the informed case,
    # a percentage will be removed dependent on the difference in the
    # uninformed and informed area conserved
    # A maximum of a 10% change in species occurrence probability is allowed.
    for br in range(0,cells):

        try:
            g.write(str([round(x-(informed[br]/uninformed[br])*0.1,4) for x in probabilities_sp(species,uninformed[br])]))
        except:
            import pdb;pdb.set_trace()
        g.write(str("\n"))
    g.write(str("];"))
    g.close()

# Setting the three values needed to run the main function, number of cells,
# number of species, and file location where data will be placed.
cells = 633
species = 170
file_name = 'C:/MyTemp/CONBIO_COMMENT/data_conbio.dat'

main(cells,species,file_name)
```

Appendix 2. CPLEX model for solving the mixed integer problem.

```
// CPLEX MODEL FOR CONDUCTING ANALYSIS ON SIMULATED DATA
// EACH RUN will create a single solution for a specific budget
// (in this case, steps of 150,000 DKK, B)
// using either uninformed (UNINFORMED =1) or informed scenario (UNINFORMED = 0) and
// using either sigmoid (SIGMOID =1) or approximate linear (SIGMOID = 0)
// survival probability functions

int UNINFORMED = 0;
int SIGMOID = 0;
int stands = ...;
int species = ...;
float pji_un[1..stands][1..species] = ...;
float pji_in[1..stands][1..species] = ...;
dvar boolean X[1..stands];
dvar float+ lambda[1..species][1..401];
dvar float s[1..species];
float B =5;
float c[1..stands] = ...;
float Bk[1..401] = ...;
float area_in[1..stands]=...;
float area_un[1..stands]=...;
dvar float+ x_sum;
dvar float+ Bud_tot;
dvar float+ sum_area;

maximize
sum(i in 1..species)(1-s[i]);

subject to {
forall(i in 1..species)
sum(k in 1..401)(ln(Bk[k])*lambda[i][k]) == SIGMOID * sum(j in 1..stands)
(X[j]*ln(1-(1-UNINFORMED)*(pji_in[j][i]*((0.829602+
(-0.006443042- 0.829602)/(1+pow((area_in[j]*10000)/1319.231,6.224721)))))-
(UNINFORMED)*(pji_un[j][i]*((0.829602+(-0.006443042-0.829602)/
(1+pow((area_in[j]*10000)/1319.231,6.224721))))))) +
(1-SIGMOID)*sum(j in 1..stands)
(X[j]*ln(1-(1-UNINFORMED)*(pji_in[j][i]*((0.9821882+
(0.02101128-0.9821882)/(1+pow((area_in[j]*10000)/2065.877,2.195059)))))-
(UNINFORMED)*(pji_un[j][i]*((0.9821882+(0.02101128-0.9821882)/
(1+pow((area_un[j]*10000)/2065.877,2.195059)))))));

forall(i in 1..species)
    s[i] == sum(k in 1..401)Bk[k]*lambda[i][k];

sum(j in 1..stands)((c[j]*((area_in[j]*10000)*(1-UNINFORMED)+
    UNINFORMED*(area_un[j]*10000)))*X[j]) <= 150000*B;

forall(i in 1..species)
   sum(k in 1..401)lambda[i][k] == 1;

x_sum == sum(j in 1..stands) X[j];

Bud_tot == sum(j in 1..stands)((c[j]*(area_in[j]*(1-UNINFORMED)+
    UNINFORMED*area_un[j]))*X[j]);

sum_area == sum(j in 1..stands)(((area_in[j]*(1-UNINFORMED)
+UNINFORMED*area_un[j]))*X[j]);
}
```