Regression Tree and Random Forest Model Validation Document

for

"A Meta-Analysis of Carbon Nanotube Pulmonary Toxicity Studies – How Physical Dimensions and Impurities Affect the Toxicity of Carbon Nanotubes"

Jeremy Gernand and Elizabeth Casman

Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, USA

6 December 2012

1 Summary

This document contains model learning statistics, and structure of the models utilized in the paper “A meta-analysis of carbon nanotube pulmonary toxicity studies – How physical dimensions and impurities affect the toxicity of carbon nanotubes.” This information is meant to supplement and support the explanations and conclusions reached in that paper.

This document includes the detailed structure of the pruned regression tree models (Figures 1 through 3) as well as the tree’s error performance as a function of model growth (Figures 4 through 7). The random forest model performance versus model growth for each of the 4 output measures is also included (Figures 8 through 11).

The stepwise random forest models and their performance as a function of included variables are displayed in Figures 12 through 15. The random forest generated dose-response profiles and the effects of cobalt impurities are shown in Section 6 (Figures 16 through 18).

The MATLAB^{TM} code that creates these regression tree and random forest model objects including the stepwise random forest model object is included in Section 7. The data used to train the models can be found at https://nanohub.org/resources/13515.

2 Pruned Regression Tree Models

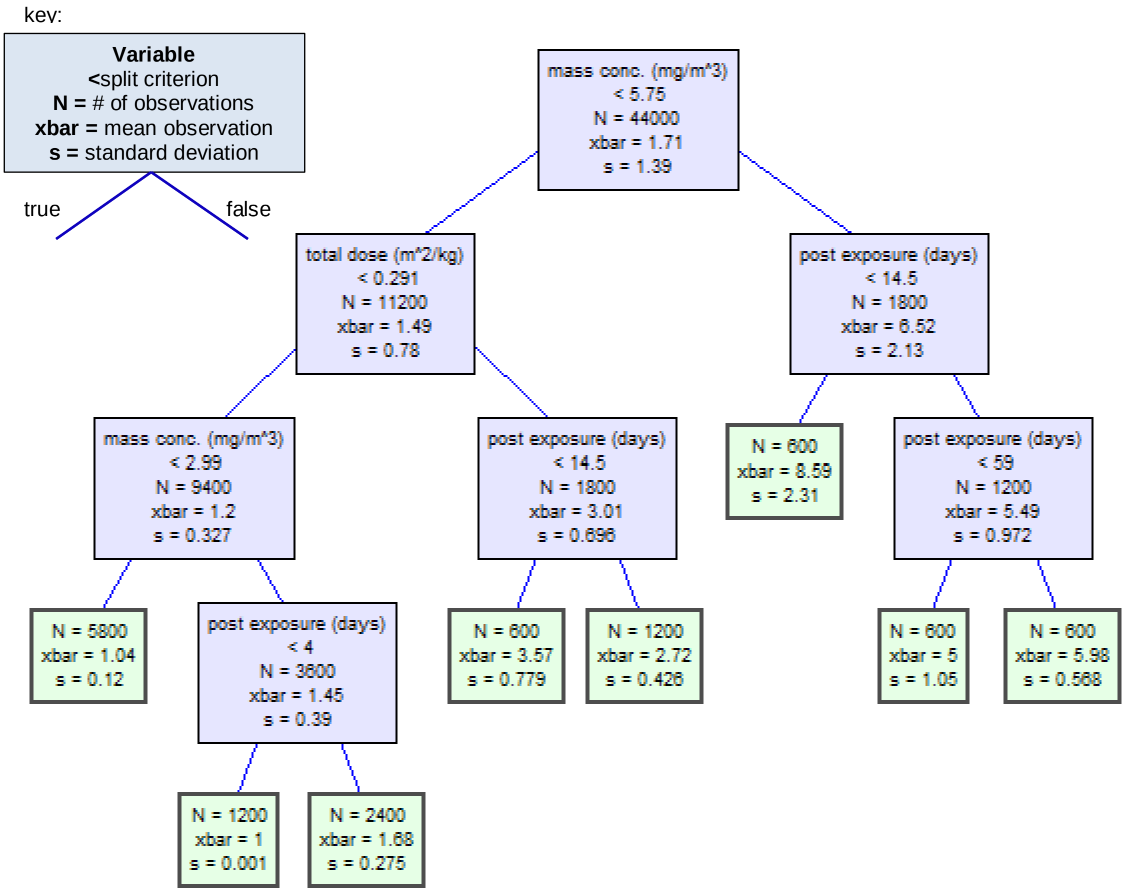

Figure 1: RT model for BAL macrophages as measured by fold of control. Each branch divides the population of observations into two child populations based on an inequality in one variable. The mean values in the leaf nodes (terminal nodes) are the model's predictions. Characteristics about the BAL macrophage values including number of observations (N), mean (xbar), and standard deviation (s) are provided at each leaf and branch.

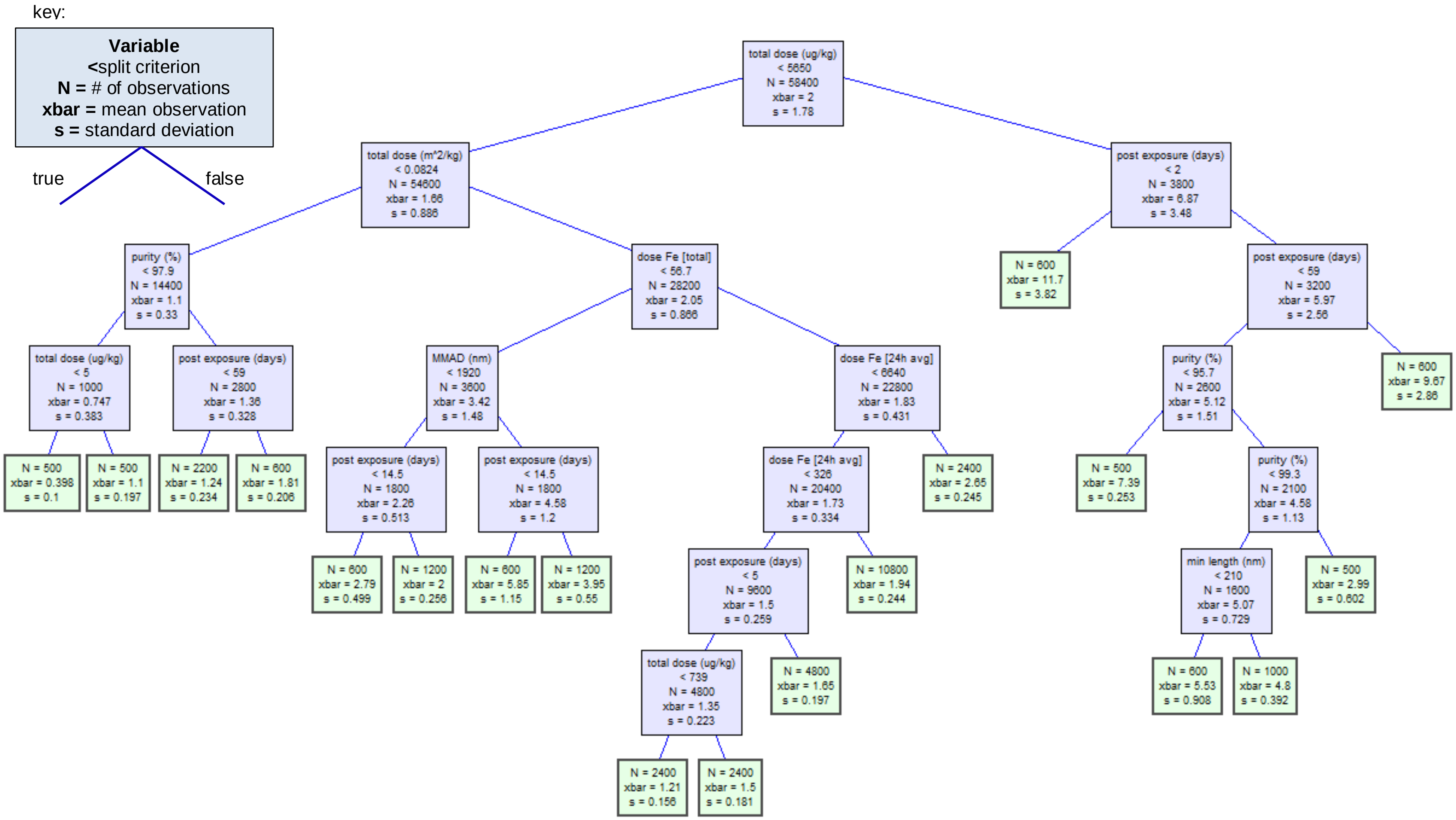

Figure 2: RT model for BAL Lactate Dehydrogenase (LDH). Each branch divides the population of observations into two child populations based on an inequality in one variable. The mean values in the leaf nodes (terminal nodes) are the model's predictions. Characteristics about the BAL LDH values including number of observations (N), mean (xbar), and standard deviation (s) are provided at each leaf and branch.

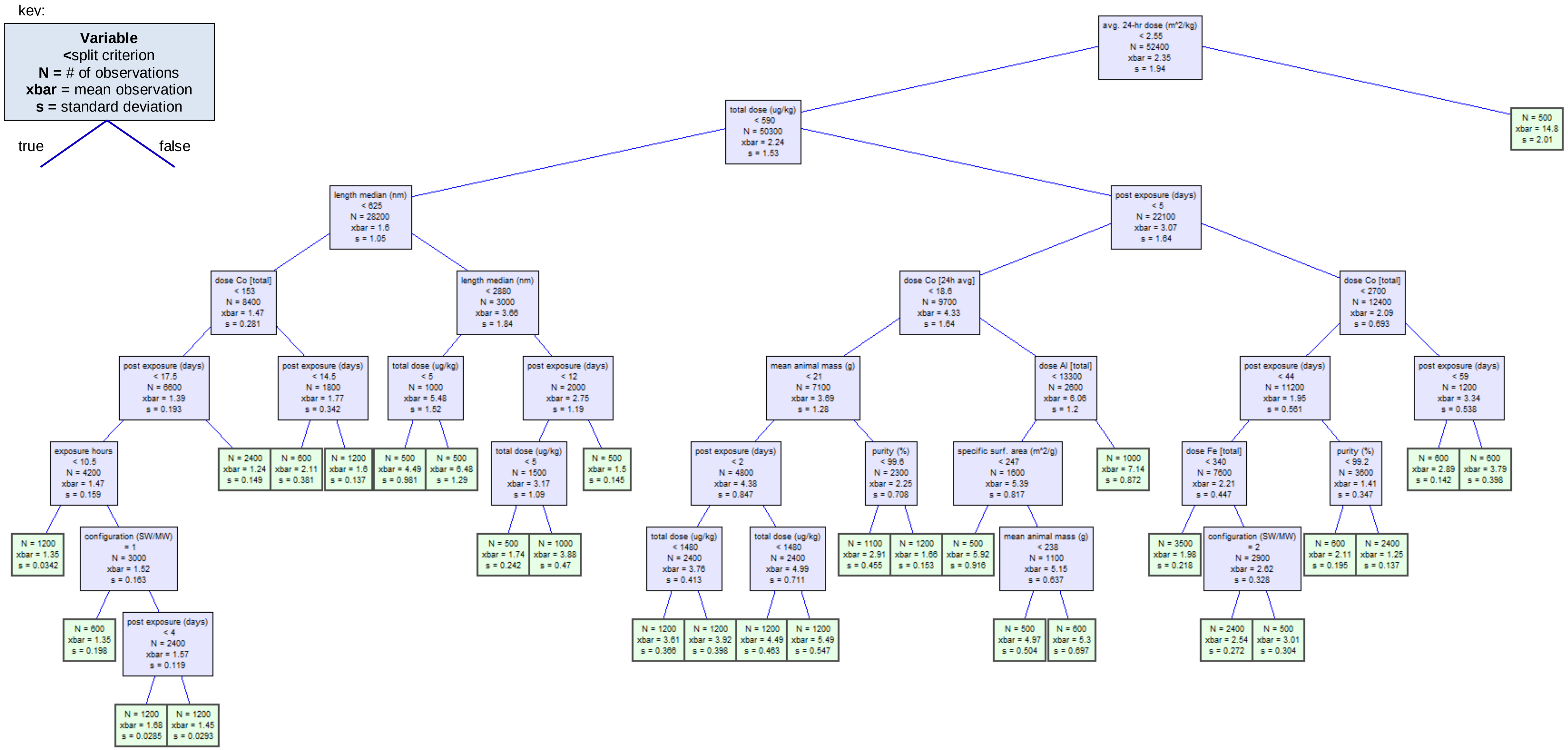
kev:
Variable
<split criterion
N = # of observations
xbar = mean observation
s = standard deviation
true
false
avg. 24-hr dose (m^2/kg) < 2.55 N = 52400 xbar = 2.35 s = 1.94
total dose (ug/kg) < 590 N = 50300 xbar = 2.24 s = 1.53
N = 500 xbar = 14.8 s = 2.01
length median (nm) < 625 N = 28200 xbar = 1.6 s = 1.05
post exposure (days) < 5 N = 22100 xbar = 3.07 s = 1.64
dose Co [total] < 153 N = 8400 xbar = 1.47 s = 0.281
length median (nm) < 2880 N = 3000 xbar = 3.66 s = 1.84
dose Co [24h avg] < 18.6 N = 9700 xbar = 4.33 s = 1.64
dose Co [total] < 2700 N = 12400 xbar = 2.09 s = 0.693
post exposure (days) < 17.5 N = 6600 xbar = 1.39 s = 0.193
post exposure (days) < 14.5 N = 1800 xbar = 1.77 s = 0.342
total dose (ug/kg) < 5 N = 1000 xbar = 5.48 s = 1.52
post exposure (days) < 12 N = 2000 xbar = 2.75 s = 1.19
mean animal mass (g) < 21 N = 7100 xbar = 3.69 s = 1.28
dose Al [total] < 13300 N = 2600 xbar = 6.06 s = 1.2
post exposure (days) < 44 N = 11200 xbar = 1.95 s = 0.561
post exposure (days) < 59 N = 1200 xbar = 3.34 s = 0.538
exposure hours < 10.5 N = 4200 xbar = 1.47 s = 0.159
N = 2400 xbar = 1.24 s = 0.149
N = 600 xbar = 2.11 s = 0.381
N = 1200 xbar = 1.6 s = 0.137
N = 500 xbar = 4.49 s = 0.981
N = 500 xbar = 6.48 s = 1.29
total dose (ug/kg) < 5 N = 1500 xbar = 3.17 s = 1.09
N = 500 xbar = 1.5 s = 0.145
post exposure (days) < 2 N = 4800 xbar = 4.38 s = 0.847
purity (%) < 99.6 N = 2300 xbar = 2.25 s = 0.708
specific surf. area (m^2/g) < 247 N = 1600 xbar = 5.39 s = 0.817
N = 1000 xbar = 7.14 s = 0.872
dose Fe [total] < 340 N = 7600 xbar = 2.21 s = 0.447
purity (%) < 99.2 N = 3600 xbar = 1.41 s = 0.347
N = 600 xbar = 2.89 s = 0.142
N = 600 xbar = 3.79 s = 0.398
N = 1200 xbar = 1.35 s = 0.0342
configuration (SW/MW) = 1 N = 3000 xbar = 1.52 s = 0.163
N = 500 xbar = 1.74 s = 0.242
N = 1000 xbar = 3.88 s = 0.47
total dose (ug/kg) < 1480 N = 2400 xbar = 3.76 s = 0.413
total dose (ug/kg) < 1480 N = 2400 xbar = 4.99 s = 0.711
N = 1100 xbar = 2.91 s = 0.455
N = 1200 xbar = 1.66 s = 0.153
N = 500 xbar = 5.92 s = 0.916
mean animal mass (g) < 238 N = 1100 xbar = 5.15 s = 0.637
N = 3500 xbar = 1.98 s = 0.218
configuration (SW/MW) = 2 N = 2900 xbar = 2.62 s = 0.328
N = 600 xbar = 2.11 s = 0.195
N = 2400 xbar = 1.25 s = 0.137
N = 600 xbar = 1.35 s = 0.198
post exposure (days) < 4 N = 2400 xbar = 1.57 s = 0.119
N = 1200 xbar = 3.61 s = 0.366
N = 1200 xbar = 3.92 s = 0.398
N = 1200 xbar = 4.49 s = 0.463
N = 1200 xbar = 5.49 s = 0.547
N = 500 xbar = 4.97 s = 0.504
N = 600 xbar = 5.3 s = 0.697
N = 2400 xbar = 2.54 s = 0.272
N = 500 xbar = 3.01 s = 0.304
N = 1200 xbar = 1.68 s = 0.0285
N = 1200 xbar = 1.45 s = 0.0293

Figure 3: RT model for BAL Total Protein. Each branch divides the population of observations into two child populations based on an inequality in one variable. The mean values in the leaf nodes (terminal nodes) are the model's predictions. Characteristics about the BAL total protein values including number of observations (N), mean (xbar), and standard deviation (s) are provided at each leaf and branch.

3 Regression Tree Variable Importance

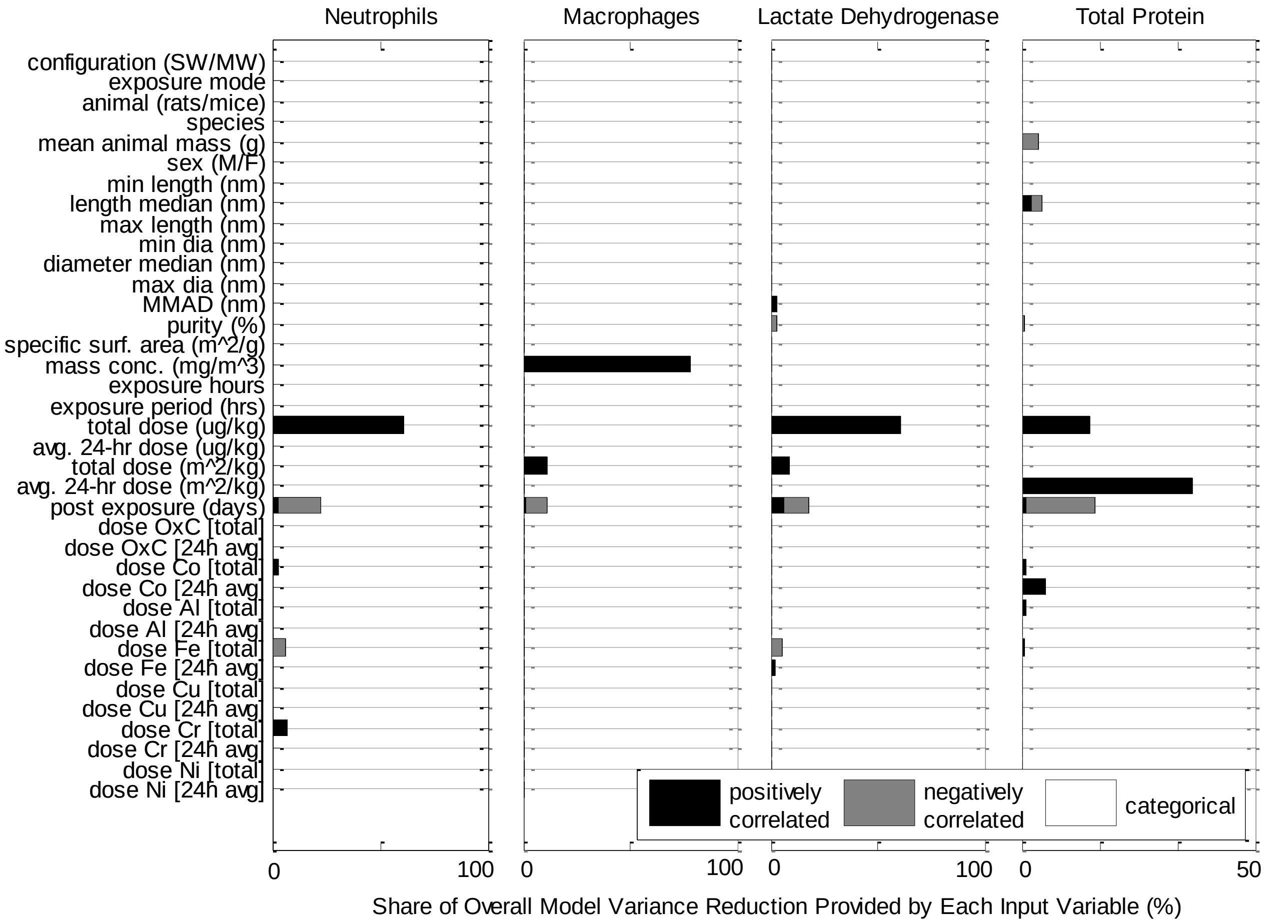

Figure 4: Variable importance as determined by the relative variance reduction for each of the 4 regression tree (RT) models. Bar shading indicates whether the variance reduction occurred with a positive or negative correlation between the input and output, or with a branch based on a non-numeric categorical variable.

4 Regression Tree Model Growth Curves

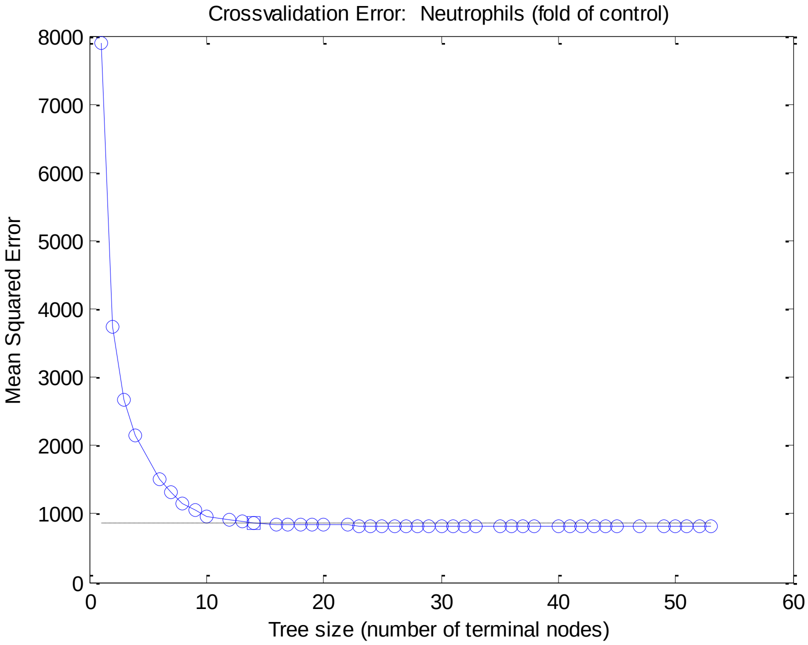

Figure 5: The regression tree model cross-validation error for BAL neutrophils. The dashed line indicates the level of one standard error above the minimum potential error. Based on these results the regression tree model is pruned to 12 branches.

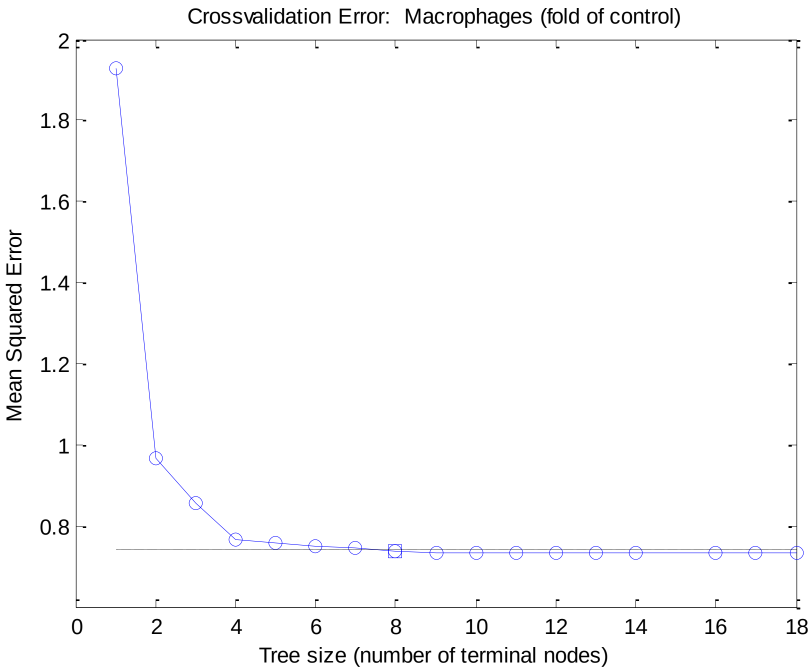

Figure 6: The regression tree model cross-validation error for BAL macrophages. The dashed line indicates the level of one standard error above the minimum potential error. Based on these results the regression tree model is pruned to 8 branches.

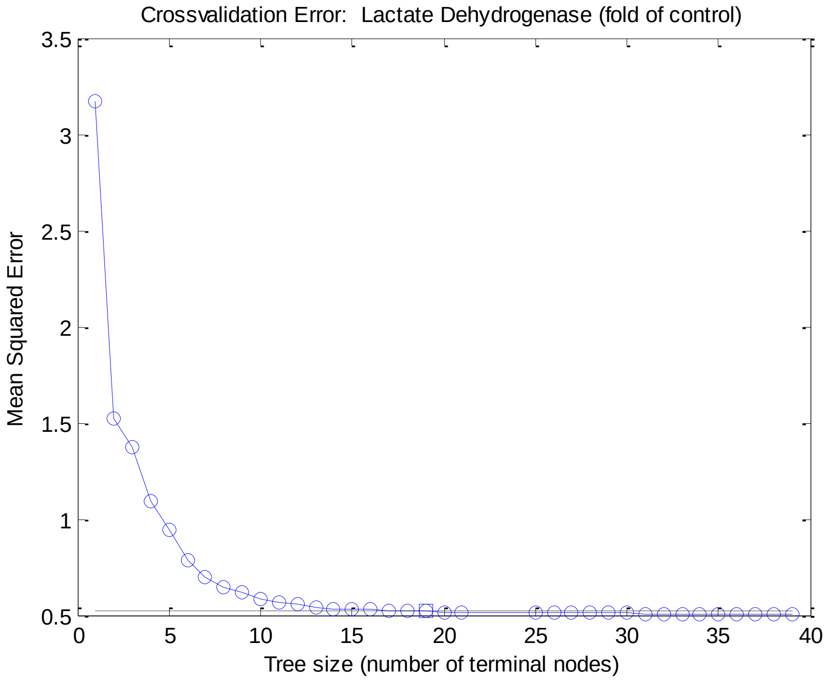

Figure 7: The regression tree model cross-validation error for BAL lactate dehydrogenase. The dashed line indicates the level of one standard error above the minimum potential error. Based on these results the regression tree model is pruned to 19 branches.

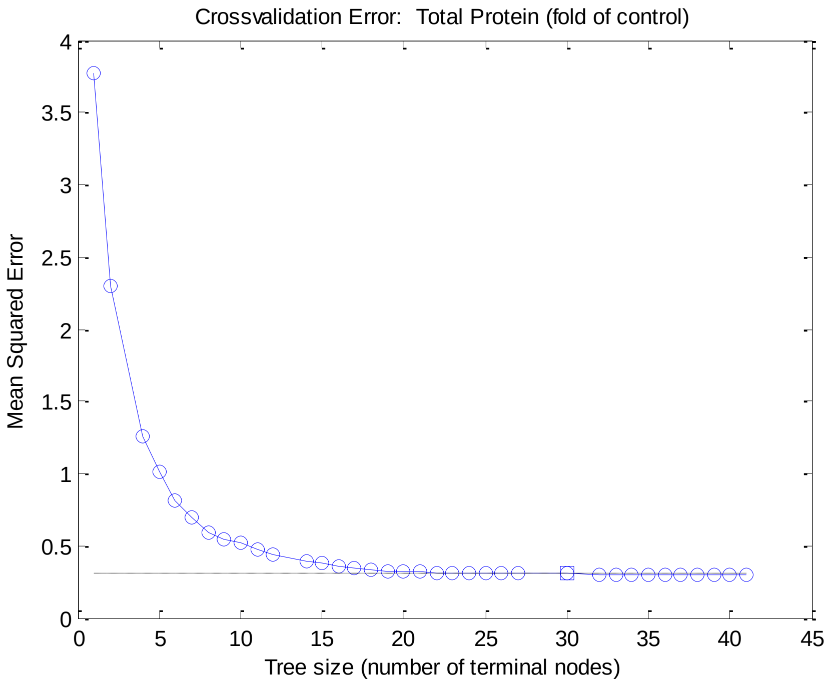

Figure 8: The regression tree model cross-validation error for BAL total protein. The dashed line indicates the level of one standard error above the minimum potential error. Based on these results the regression tree model is pruned to 30 branches.

5 Random Forest Model Growth Curves

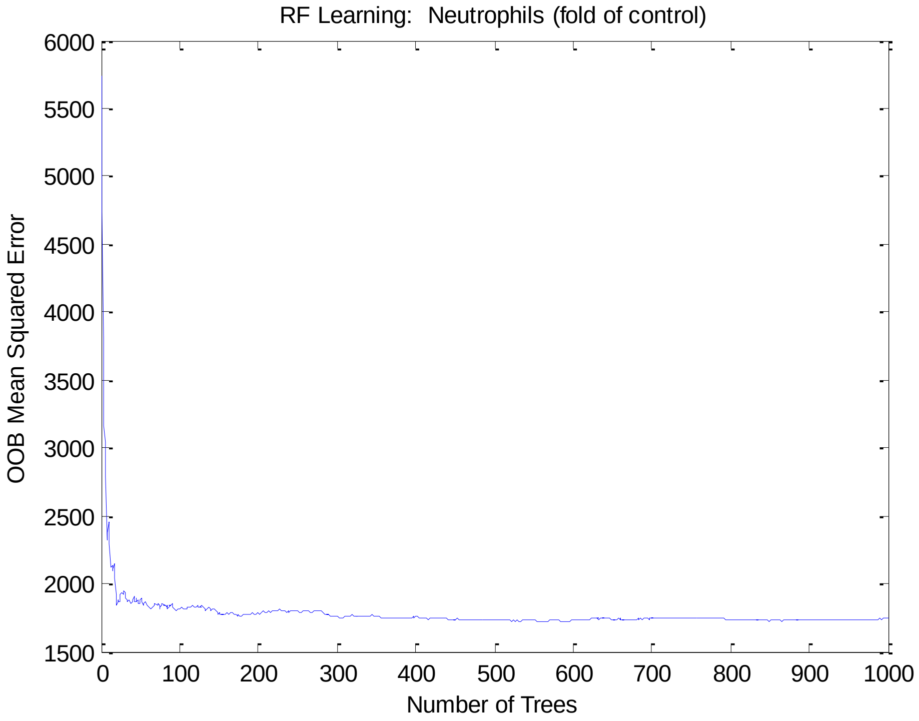

Figure 9: The random forest model out-of-bag (out-of-bag describes data samples withheld from the training of individual tree models making up the forest) mean squared error as a function of tree models included in forest for BAL neutrophils. These results reflect that nearly all potential information gain has been achieved prior to 1000 trees being included.

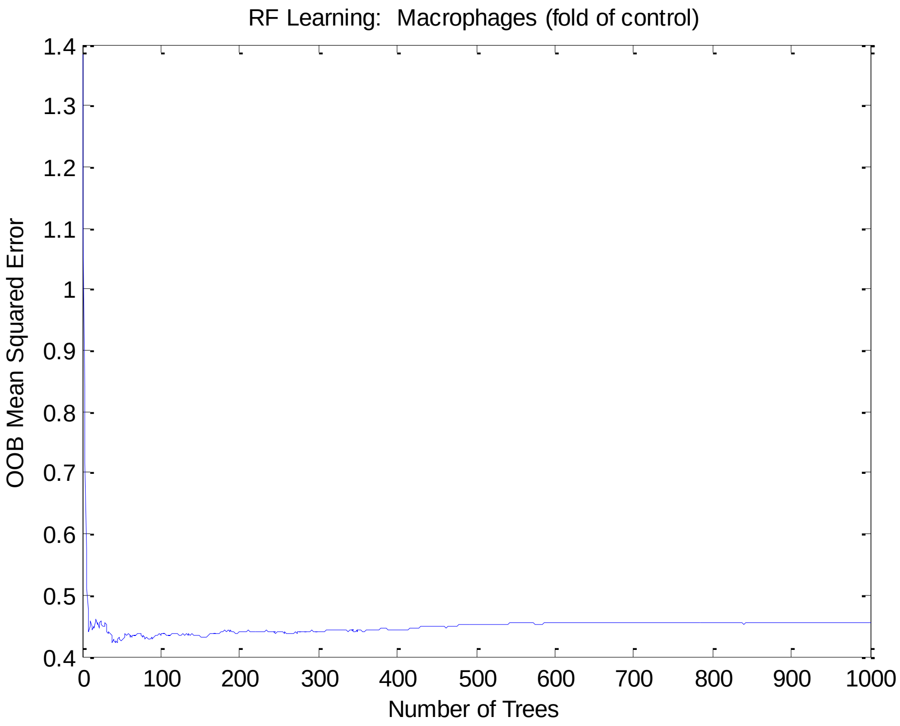

Figure 10: The random forest model out-of-bag (out-of-bag describes data samples withheld from the training of individual tree models making up the forest) mean squared error as a function of tree models included in forest for BAL macrophages. These results reflect that nearly all potential information gain has been achieved prior to 1000 trees being included.

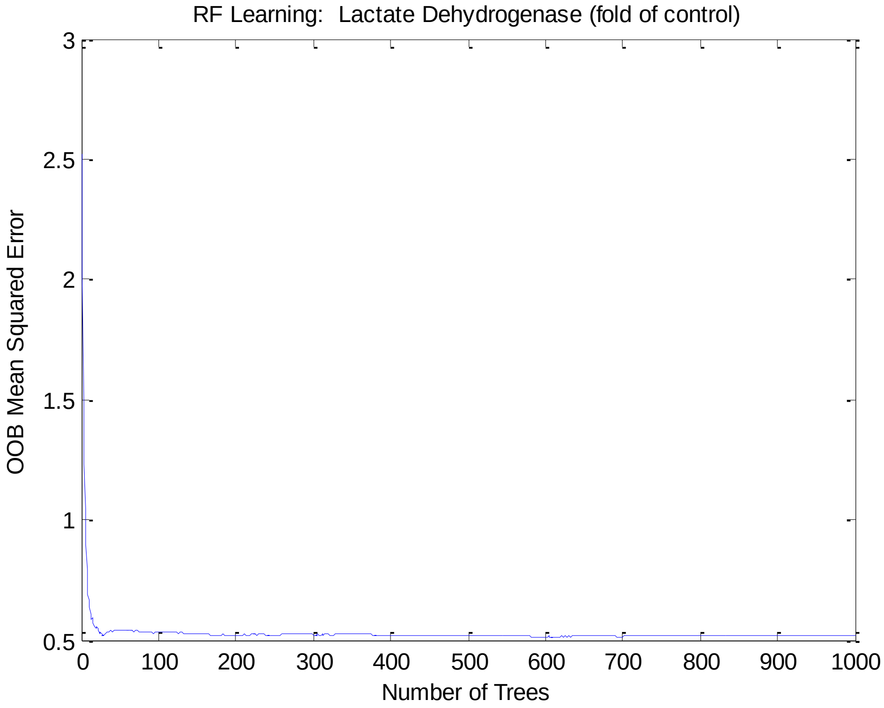

Figure 11: The random forest model out-of-bag (out-of-bag describes data samples withheld from the training of individual tree models making up the forest) mean squared error as a function of tree models included in forest for BAL lactate dehydrogenase. These results reflect that nearly all potential information gain has been achieved prior to 1000 trees being included.

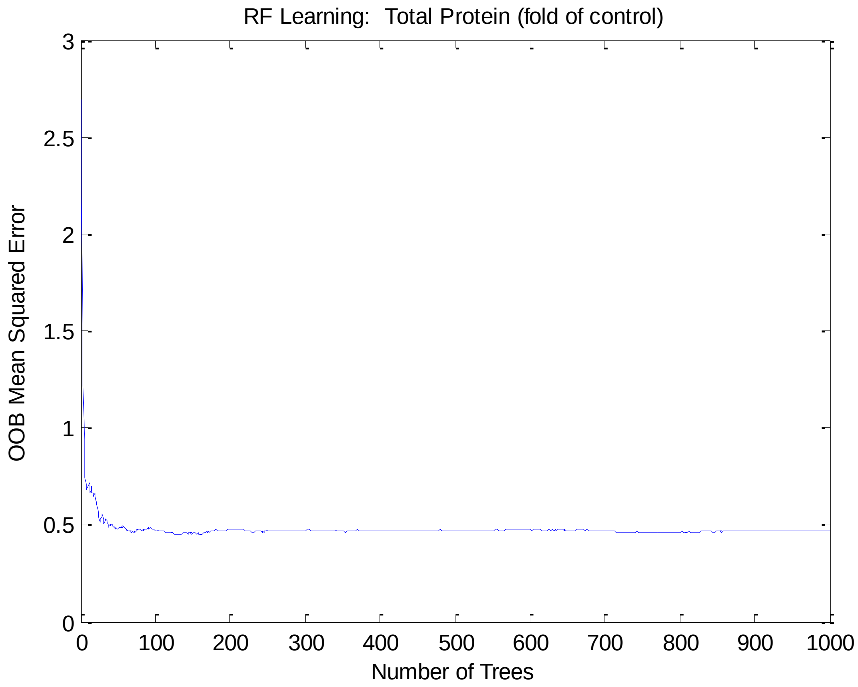

Figure 12: The random forest model out-of-bag (out-of-bag describes data samples withheld from the training of individual tree models making up the forest) mean squared error as a function of tree models included in forest for BAL total protein. These results reflect that nearly all potential information gain has been achieved prior to 1000 trees being included.

6 Stepwise Random Forest Model Growth Curves

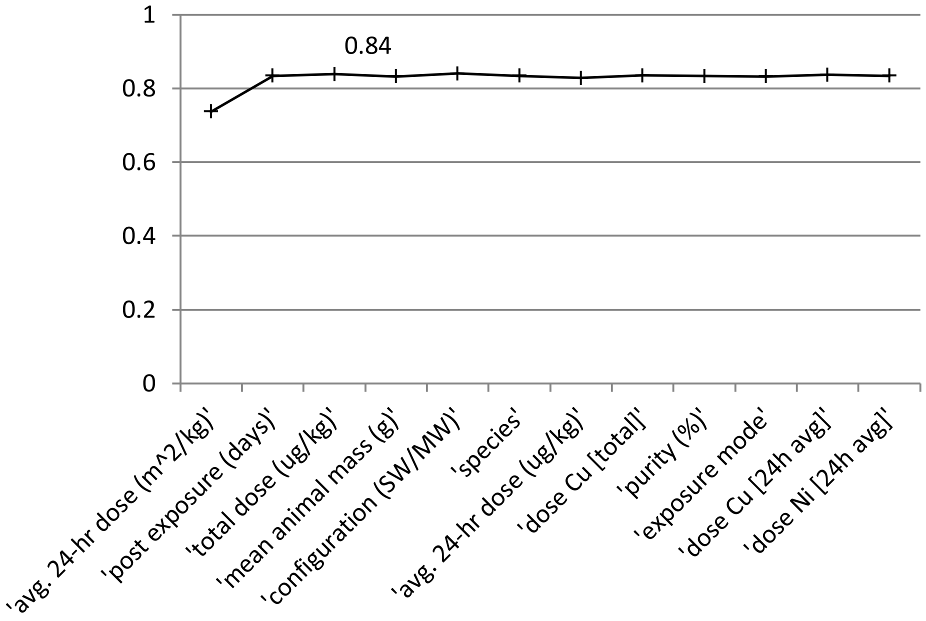

Figure 13: Stepwise RF model performance by added variable for BAL macrophages. The maximum performance of 0.84 is reached at the addition of the 3rd model parameter, total dose.

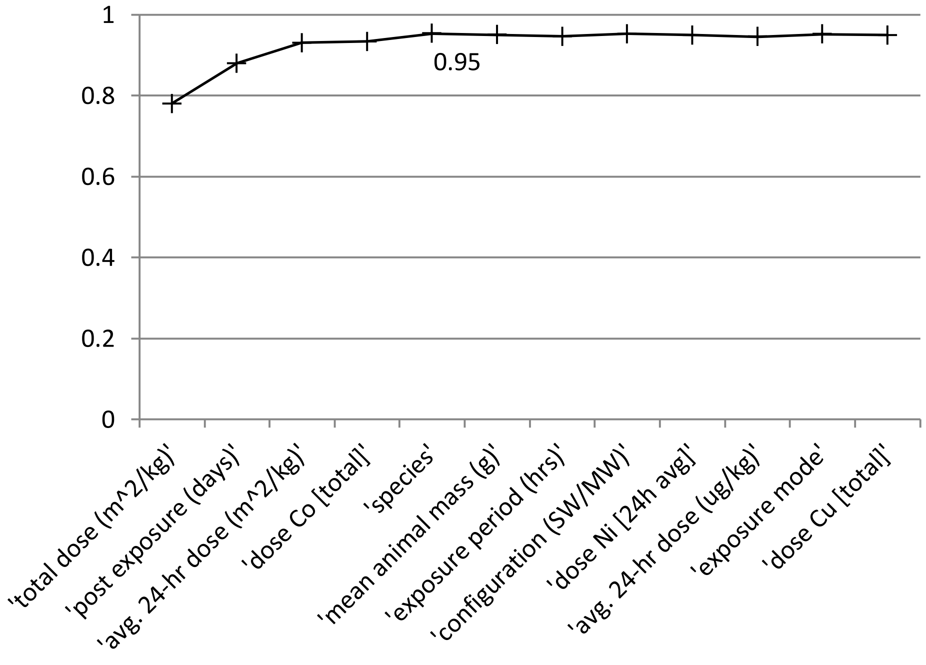

Figure 14 Stepwise RF model performance by added variable for BAL Total Protein. The maximum performance of 0.95 is reached at the addition of the 5th model parameter, species.

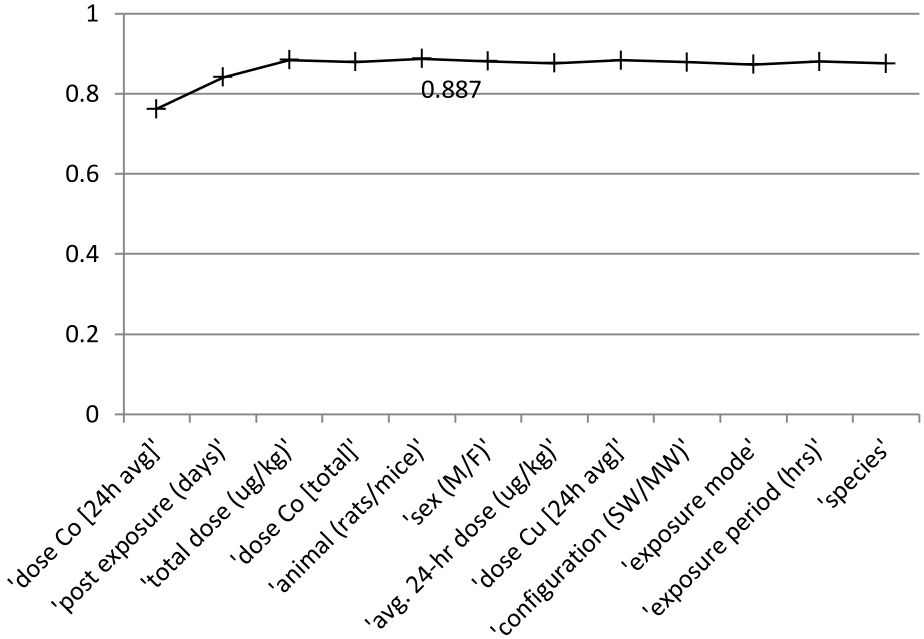

Figure 15: Stepwise RF model performance by added variable for BAL Lactate Dehydrogenase (LDH). The maximum performance of 0.89 is reached at the addition of the 5th model parameter, animal type.

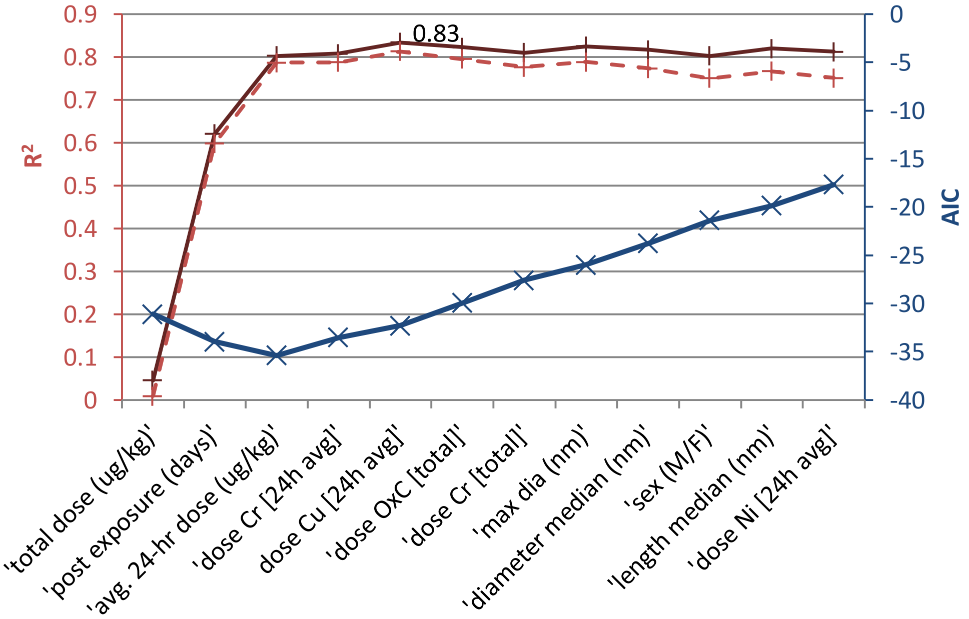

Figure 16: Stepwise RF model performance by added variable for BAL neutrophils. The maximum performance of 0.83 is reached at the addition of the 5th model parameter, 24 hour average dose of copper. The Akaike Information Criterion (AIC) is plotted on the right axis.

7 Random Forest Model Dose-Response Curves and Cobalt Impurity Impact

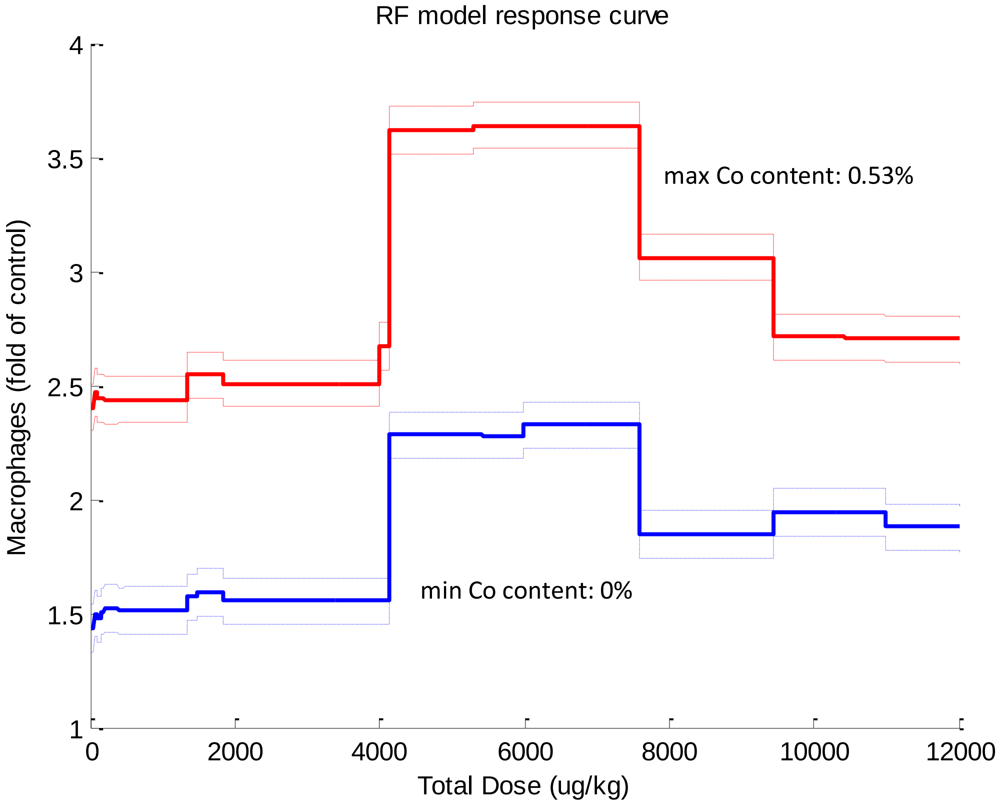

Figure 17: RF model dose-response response curve for macrophages as modified by cobalt content. Dashed and dotted lines indicate the RT model predicted experimental standard deviation. All other model inputs are held constant at their median value.

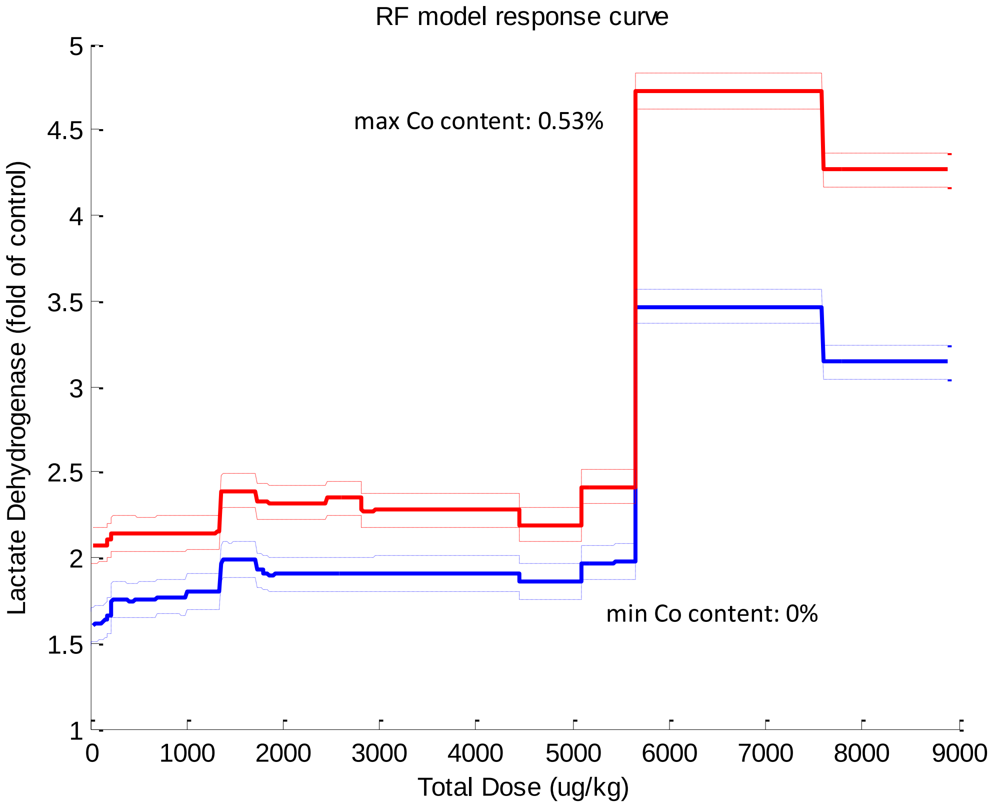

Figure 18: RF model dose-response response curve for lactate dehydrogenase as modified by cobalt content. Dashed and dotted lines indicate the RT model predicted experimental standard deviation. All other model inputs are held constant at their median value.

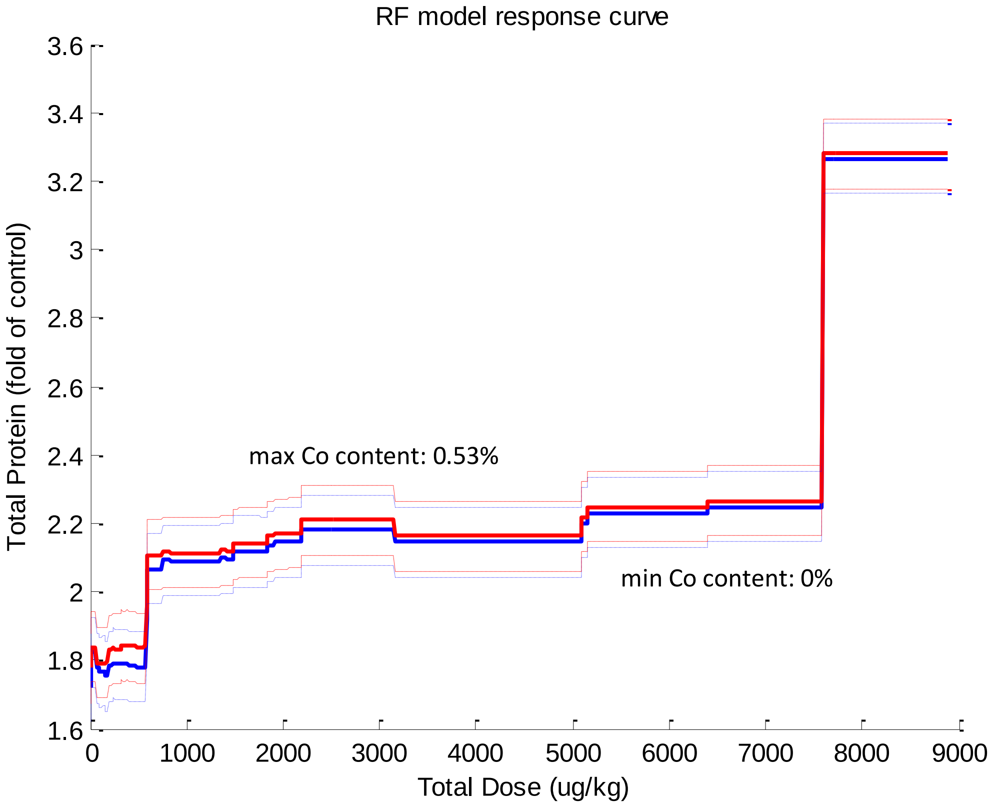

Figure 19: RF model dose-response response curve for BAL total protein as modified by cobalt content. Dashed and dotted lines indicate the RT model predicted experimental standard deviation. All other model inputs are held constant at their median value.

8 MATLAB Code for Regression Tree and Random Forest Model Generation

Regression Tree Model:

```
catcol = [1 2 3 4 6];
%numerical designations of 'Inputs' columns that are categorical

t = classregtree(Inputs,Outputs,'names',InputNames,'categorical',catcol);

[crscost,crserr,crsnodes,crsbstlvl] = ...
    test(t,'crossvalidate',InputsNew,OutputsNew,'nsamples',10);

tmin = prune(t,'level',crsbstlvl);               %crsbstlvl for auto pruning


fig(1) = view(tmin,outputcol,OutputNames{outputcol});



[mincrscost,mincrsloc] = min(crscost);
figure('Name',['Crossvalidation Error:  ',OutputNames{outputcol}]);
    plot(crsnodes,crscost,'b-o',...
    crsnodes(crsbstlvl+1),crscost(crsbstlvl+1),'bs',...
    crsnodes,(mincrscost+crserr(mincrsloc))*ones(size(crsnodes)),'k--');
    xlabel('Tree size (number of terminal nodes)');
    ylabel('Mean Squared Error');
    title(['Crossvalidation Error:  ',OutputNames{outputcol}]);
```

Random Forest Model:

```
numtrees = 1000;     %numer of trees in forest
catcol = [1 2 3 4 6];        %numerical designations of 'Inputs' columns that
are categorical

inputcol = size(Inputs,2);

b = TreeBagger(numtrees,Inputs,Outputs,'method','r',...
      'oobpred','on','oobvarimp','on','NVarToSample',round(inputcol/3),...
      'categorical',catcol);

figure('Name',['RF Learning:  ',OutputNames{outputcol}]);  % Plot of error as
a function of trees in model
    plot(oobError(b));
    xlabel('Number of Trees');
    ylabel('OOB Mean Squared Error');
    title(['RF Learning:  ',OutputNames{outputcol}]);
```

Stepwise Random Forest Model:

```
%   This script creates a stepwise random forest model.
%
%   24 January 2012
%   (c) Jeremy M. Gernand
%
%

first = tic;        %start CPU timer
disp('   Initializing...')
%   Designate output variable and process parameters **********************
%
%   >>>>>>>>>>>>>>>>>>>>>>>>>>> USER INPUT <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
outputcol = 23;      %designate output column
numtrees = 1000;            %numer of trees in forest
catcol = [1 2 3 4 6];       %numerical designations of 'Inputs' columns
                            %   that are categorical
%   >>>>>>>>>>>>>>>>>>>>>>>>>>> END USER INPUT <<<<<<<<<<<<<<<<<<<<<<<<<<<<

%   Capture matrix parameters **********************************************
%
inputcol = size(Inputs,2);

%   Trim inputs and outputs to only those with numerical output value *****
%
InputsNew = Inputs;
OutputsNew = Outputs(:,outputcol);
for i = 1:size(Inputs,1);
    if isnan(OutputsNew(size(Inputs,1)-i+1));

        OutputsNew(size(Inputs,1)-i+1) = [];
        InputsNew(size(Inputs,1)-i+1,:) = [];

    end
end

%   Set up and record results of stepwise random forest models ************
%
disp('   Generating stepwise random forest model...')

DataArray = [];
NewDataArray = [];
UsedVars = [];
CatCols = [];
OutputMean = mean(OutputsNew);
rfTSS = 0;
er_count = size(InputsNew,1);
TSSarray = OutputsNew(1:er_count) - OutputMean;
TSSarray = TSSarray.^2;
rfTSS = sum(TSSarray);
RFstepwiseR2 = zeros(round(inputcol/3),inputcol);
RFstepwiseStats = cell(round(inputcol/3),4);
```

Continued…

```
for step = 1:round(inputcol/3);
    second = tic;

    %   display progress on screen
    stepstr = ['      step ',num2str(step),'... '];
    disp(stepstr)

    %   define data array for RF model generate (loop)
    for trial = 1:inputcol;
        %   check if variable has already been used (skip calc if true)
        if ~isempty(find(UsedVars==trial,1));
            RFstepwiseR2(step,trial) = NaN;
        else

            %   define new total data array
            NewDataArray = InputsNew(:,trial);
            InputArray = [DataArray NewDataArray];

            %   set up list of categorical variables
            if ~isempty(find(catcol,trial));
                CatCols = [CatCols trial];
            end

            NumColInMatrix = size(InputArray,2);

            %   define NumVarsToSample
            if NumColInMatrix == 1;
                NumVarsToSample = 1;
            elseif NumColInMatrix == 2;
                NumVarsToSample = 1;
            elseif NumColInMatrix > 2;
                NumVarsToSample = round(NumColInMatrix/3);
            end

            %   generate RF model and compact
            trialstr = ['          trial ',num2str(trial),'... '];
            disp(trialstr)

            b = TreeBagger(numtrees,InputArray,OutputsNew,'method','r',...

'oobpred','on','oobvarimp','on','NVarToSample',NumVarsToSample,...
                'categorical',CatCols);
            bcompact = compact(b);

            %   calculate and record R-squared value for RF model
            rfSSE = 0;
            [RFpredict,RFsd] = predict(bcompact,InputArray);
            SSEarray = OutputsNew(1:er_count) - RFpredict(1:er_count);
            SSEarray = SSEarray.^2;
            rfSSE = sum(SSEarray);
            RFstepwiseR2(step,trial) = 1 - (rfSSE / rfTSS);

        end

    end
```

Continued…

```
%   select highest performing variable
    [BestTrial, BestLocation] = max(RFstepwiseR2(step,:));

    %   record elapsed time
    elapsedtime(step) = round(toc(second)/60);

    %   record performance stats
    RFstepwiseStats(step,1) = {step};               %stepwise row number
    RFstepwiseStats(step,2) = {BestLocation};       %variable number
    RFstepwiseStats(step,3) = InputNames(BestLocation);     %variable name
    RFstepwiseStats(step,4) = {RFstepwiseR2(step,BestLocation)}; %R^2 value
    RFstepwiseStats(step,5) = {elapsedtime(step)};          %step time(min)

    %   define new input array
    DataArray = [DataArray InputsNew(:,BestLocation)];
    UsedVars = [UsedVars BestLocation];

    %   display progress on screen
    progstr = ['       R2 progress ',num2str(BestTrial),'... '];
    disp(progstr)
    timestr = ['       time elapsed (min): ',num2str(elapsedtime(step))];
    disp(timestr)

end

%   Graph plot of R-squared versus variable addition
%       include order of inclusion on graph...

figure('Name',['RF Stepwise Model Growth:  ',OutputNames{outputcol}]);
hold on;
R2fig = plot(RFstepwiseStats{:,1},RFstepwiseStats{:,4},'-');
title(['RF Stepwise Model Growth:  ',OutputNames{outputcol}]);
xlabel('Variable Names in Order of Addition');
ylabel('Model Performance (R^2)');
hold off;
        %here, need to add axis labels, put variable names on x-axis
        %change formatting to show point markers with line

toc(first);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

9 Stepwise Linear Regression Models

Table 1: Coefficients and model performance statistics for stepwise linear regression models. These results provide another perspective on input variable importance, however the amount of data excluded from these models reduces the confidence as compared to the RT and RF models.

Output Variable	Input Variable	Coefficient	P-Value
Neutrophils	Mass Concentration	170.09	0
	Post Exposure Period	-38.75	5.23E-175
	MMAD	-18.86	1.69E-9
R^2 = 0.66 86% of the records excluded from this model due to missing values			
Macrophages	Mass Concentration	2.47	0
	Configuration (SW/MW)	0.91	1.10E-39
	MMAD	-0.55	3.62E-4
	Post Exposure Period	-0.20	5.10E-9
R^2 = 0.80 75% of the records excluded from this model due to missing values			
Lactate Dehydrogenase	MMAD	3.94	2.37E-47
	Configuration (SW/MW)	3.36	9.02E-166
	Mass Concentration	2.47	0
	Post Exposure Period	-0.21	7.12E-33
R^2 = 0.71 82% of the record excluded from this model due to missing values			
Total Protein	Mean Animal Mass	-14.84	1.62E-95
	Configuration (SW/MW)	8.56	1.49E-81
	MMAD	2.73	3.75E-55
	24h Avg Dose Cobalt	1.90	3.07E-95
	Mass Concentration	-1.16	5.57E-42
	Post Exposure Period	-0.31	1.23E-294
R^2 = 0.66 74% of the records excluded from this model due to missing values			