OCCAM
A Reconstructability Analysis Program

(Organizational Complexity Computation and Modeling)

Kenneth Willett and Martin Zwick

Systems Science Ph.D. Program
Portland State University, Portland OR 97207

This manual was last revised on: May 9, 2003

Occam version 3.2.7

Table of Contents

I. OVERVIEW OF RECONSTRUCTABILITY ANALYSIS

For papers on Reconstructability Analysis, see the Discrete Multivariate Modeling page at http://www.sysc.pdx.edu/res_struct.html . For an overview of RA, see the following two papers which are on the DMM page:

> "Wholes and Parts in General Systems Methodology" at
> http://www.sysc.pdx.edu/download/papers/wholesg.pdf
>
> "An Overview of Reconstructability Analysis" at
> http://www.sysc.pdx.edu/download/papers/ldlpitf.pdf

II. ACCESSING OCCAM

OCCAM LOCATION & GENERAL USE

Occam3 is at http://dmm.sysc.pdx.edu/occam.It can also be accessed from the DMM web page: http://www.sysc.pdx.edu/res_struct.html .

Occam runs on a PSU server. The user uploads a data file to this server, provides additional input information on a web input page, and then initiates Occam action. When the computation is complete, Occam either returns .html output directly to the user or a .csv output file which can be read by a spreadsheet program such as Excel. If the computation is not likely to finish rapidly, the user can provide his/her email address and Occam will email the output (in .csv form) when it is done.

NOTIFY US OF PROGRAM BUGS & MANUAL OBSCURITIES/ERRORS

If you encounter any bugs or mysterious output, please check to see that your input file matches the format requirements specified below. If you are confident that your input file is OK, email it and a description of the anomalous behavior (or the full output produced) to Ken Willett, kwillett@ignisys.com.

We also need your support in maintaining this user's manual. Please let us know if there is information missing in this manual which you need, if explanations are obscure, or if there are any errors. Email your comments to Martin Zwick, zwickm@pdx.edu.

ACTION

When one brings Occam up, one first must choose between two Occam "actions": "Do Fit" and "Do Search". There is also a "Show Log" option, which allows the use to track the status of jobs submitted for background processing. You can see this first web page by clicking on

http://dmm.sysc.pdx.edu/occam/

After selecting one of these, Occam returns a window specific to the choice made. Search assesses *many models* either from the full set of all possible models or from various partial subsets of models. Fit examines *one model* in greater detail. In an exploratory mode, one would do Search first, and then Fit, but in a confirmatory mode, one would simply do Fit. Let's focus first on the main option of "Do Search."

III. SEARCH INPUT

The Occam search input page can be viewed at:

http://dmm.sysc.pdx.edu/occam/weboccam.cgi?action=search

On the first line, the user must specify a data file, which not only provides the data to be analysed but also describes the variables used and allows the user to set certain parameters. After the data file will now be discussed, the other parameters on this input page will be explained.

DATA FILE

The user must specify a data file on the user's computer by typing its name (and location) in or finding it by browsing. The data file is then uploaded to the Occam server. This is actually all that is needed to submit an Occam job, if the user is satisfied with the default setting of all the parameters.

Data files are ascii files, generated by Notepad, or Word (where file is saved in .txt format), or Excel (where file is saved in .prn format, i.e., with space-separated columns). A minimal data file looks like this (this is the data from the Wholes & Parts paper):

```
:action
search

:nominal
alpha,      2,1,a
beta,       2,1,b
gamma,      2,2,c

:data
 0    0    0    143
 0    0    1    253
 0    1    0     77
 0    1    1    182
 1    0    0    227
 1    0    1    411
 1    1    0     46
 1    1    1    139
```

This simple file has 3 parts: (1) the action specification, (2) specification of the variables, (3) the data to be analyzed (the data is a sequence of rows ["records"] which specify values for all the variables.) Each part in this example begins with a line of the form ":parameter", where "parameter" is "action", "nominal", or "data".

Variable specification

The action specification is straightforward. Variable specification begins with ":nominal" which reminds the user that nominal (categorical, qualitative) variables must be used. In the future, Occam will be able to bin (discretize) values of quantitative variables and thus convert the variables to nominal, but at present, the user must do this conversion. (Unless there are compelling reasons to choose a different number of bins, 3 bins is a good default; this allows detection of nonlinearities.) After ":nominal", the variables are specified, one per line. In the above example, the first line is (spaces in these lines are ignored):

alpha, 2,1,a

"alpha" is the name of the first variable. The second field indicates that it has 2 possible states (a "cardinality" of 2). The third field (shown above as 1) is either 0,1 or 2. 1 means the variable is an "independent variable" (IV) or input. 2 means it is a "dependent variable" (DV) or output. A value of 0 means that the variable (and the corresponding column in the data) will be ignored. This allows the user to have data for more variables than can be analyzed at any one time; the user could then easily alter which variables are to be included in the analysis and which are to be omitted.The value of 0 in the third field also supersedes any rebinning specification (described below); the rebinning string will be completely ignored if the third field is 0.

The fourth field is a variable abbreviation, usually one letter, in lower case, but used in Occam output in upper case. In the above example, alpha will be referred to in Occam output as A. If there are more than 26 variables, one can use double (or triple, etc.) letters as abbreviations, for example "aa" or "ab". Such a variable will appear in a model name with only its first letter capitalized, e.g., AaB:AbC:… Variable abbreviations *must* be only (one or more) letters; numbers or other symbols may not be used to abbreviate variables.

If all variables are designated as IVs (1) or as DVs (2), the system is "neutral." If some variables are IVs, and at least one is a DV, the system is "directed." The above data file is for a directed system.

Rebinning (Recoding) Feature

This feature allows the user to

(a) *ignore* data where some variables have particular values,
(b) *select* only data where some variables have particular values, and
(c) *regroup* (recode) states of a variable.

(By default this feature is turned ON. If you are not actually using this feature, it being on will only add very slightly to the time of a run, but to turn this feature OFF say ": no-rebin" anywhere before ": nominal" in the data file. This makes Occam deactivate the rebinning module and if rebinning parameters are specified in the variable specification

Occam ignores them. Also if the variable is marked to be ignored – the third field in the variable specification being 0 -- then any rebin string which follows is ignored.)
There is a simple way that one can *ignore* or *select* a single state of a variable. It involves adding a 5th field, as follows. Ignoring a state is done as follows:

Alpha, 4,1,a,exclude(1)

will exclude state 1 information of Variable Alpha from the analysis; that is, all data having Alpha = 1 will not be considered. The motivation for this might be that for some cases (records) values may be missing for some variables; or, one might want to exclude outliers or other particular values. In SPSS, missing data is marked by the character".", and this convention may be used in the data given to Occam (see Data Specification, below). Thus, to exclude records in which Alpha is missing, the 5th field would be "exclude(.)".

By contrast, Alpha, 4,1,a,1

has the reverse effect: only data where Alpha = 1 will be considered for analysis. Also since Alpha has only one state for analysis, variable Alpha will be lost.

One can also *regroup* several values of a variable into a new value. One might want to do this if the variables were originally binned with unnecessarily too many bins, or one wishes to reduce the number of bins for one variable to be able to use more bins for another variable or use more variables. For any given sample size the statistical significance of a result will depend on the product of the number of bins of all variables considered.

Regrouping is done by specifying a fifth field in a variable definition surrounded by brackets, and having no spaces between any of the characters inside the brackets (the rebinning string is *"white space intolerant"*). For example

theta, 3,1,t, [1(1,2);2(3)]

In this example, theta originally has 3 states but because of rebinning, old states 1 and 2 now become new state 1 and old state 3 becomes new state 2. The cardinality of theta has become 2. The general form of this regrouping specification is

[new_state (old_state , old_state……..) ; new_state (old_state,…….);……]

An old state cannot be present in more than 1 bin. Note the commas between old states and the semi-colons between new states.

Regrouping can also be used to select or ignore *more than one* state of a variable.

Some uses of Regrouping

1) To *ignore* more than one state of a variable:

Alpha, 4,1,a,[1(1),2(2)]

Values 3 and 4 of Alpha are excluded, that is all data records (rows) having such Alpha values are omitted from the analysis. If one uses this approach to exclude a single state, the result is equivalent to using "exclude( )" as the 5th field.

2) To *select* more than one state of a variable, and (thus in effect) to omit the variable

Alpha, 4,1,a,[1(1,2)]

Only data entries (rows) with Alpha equals 1 or 2 are considered; data entries with Alpha equals 3 and 4 are ignored. Variable Alpha is thus lost (the column for Alpha is ignored). The motivation for this usage is that one wishes to do the analysis of other variables only for particular values of the specified variable(s).

3) To *regroup* states, i.e., to reduce the number of states of a variable (this also includes non-sequential states).

Alpha, 4,1,a,[1(1,3);2(2,4)]

The cardinality of A changes from 4 to 2.

4) To combine ignoring and regrouping.

Alpha, 4,1,a,[1(1,3);2(2)]

causes data where Alpha = 4 to be ignored; also old states 1 and 3 become new state 1. The cardinality of Alpha becomes 2.

Finally, there is a wild card character that rebinning module identifies which is "*", which means "everything else." This can be used only in the last bin as in

kappa, 5,1,k, [1(1,3);2(4);3(*)]

In this case kappa will be rebined and original states 1 and 3 will become new state 1, original state 4 will become new state 2 and rest of the states of kappa will become new state 3 (in this case states 2 and 5).

Data specification

The third part of this file is the data, which follows the ":data" line. In the data, variables are columns, separated by one or more spaces. The columns from left to right correspond to the sequence of variables specified above, i.e., the first column is alpha, the second beta, and the third gamma. Following the variable columns there can be an additional column which gives the frequency of occurrence of the particular state

specifed by the variable values. The frequency value does not have to be integer, so frequencies which become non-integer because some weighting has been applied to them are OK. But frequency values may not be negative.

Note that since non-integer frequencies are allowed, one can use Occam to analyse – and compress -- *arbitrary functions* of nominal variables. Occam just scales the function value so that it can be treated as a probability value, and then does a decompositional analysis on this "probability distribution." In the RA work of Bush Jones, this is called "g-to-k" normalization. Note, however, that if Occam is used in this way, statistical measures like alpha do not have their usual interpretation.

Since variables are nominal, their values (states) are *names*. Normally, these will be 0,1,2,… or 1,2,3,….but the character "." is also allowed, e.g., to designate missing values. *No other non-numeric characters are allowed as variable states*. To avoid possible confusion, it is best to start the labelling of *all* variables either with 0 or with 1, i.e., avoid starting one variable with 0 and another with 1 (though Occam can handle such inconsistencies of convention). The user should know the number of different states that occur for each variable and indicate the cardinality of the variable correctly in the variable specification.

Data can be provided to Occam *without frequencies*, where each line (row) represents a single *case*. The rows do not have to be ordered in any particular way. Occam will generate the frequencies itself, but it needs to be told that the data do not include frequencies, as follows:

```
:no-frequency

:data
 0   0   0
 1   1   1
 0   1   1
 1   0   1
 0   0   1
 0   1   0
 0   1   1
 0   0   1
 1   0   0
 1   0   1
 1   1   0
 1   1   1
```

Uploading data will be faster if the data provides frequencies, so if the data file is big, the user might consider doing this operation before calling Occam.

Missing values

In the data that Occam actually sees, a row (case) and column (variable) cannot have a missing value (a blank in a variable's field). In preparing data for Occam, a missing value can be handled in one of three ways: (a) the row can be deleted from the data, (b) an additional value for the variable can be defined, which means "missing" (for example, if the variable is binary with states 0 and 1, a missing value could be assigned a new value of 2 and the cardinality of the variable would become 3), or (c) the value can be assigned randomly according to the observed probabilities of the different values in the rest of the data (this must be done by the user before running Occam). If only a few rows have missing values, (a) is the best choice. Note that the rebinning option described above allows one to have Occam omit rows (cases) where variables are marked as having missing values.

Additional parameter specifications

In addition to action, variables, and data, the data file may include additional parameter specifications. A parameter specification is either just a single line when the parameter is a "switch," such as the "no-frequency" parameter shown above, or it involves two lines, the first giving the parameter name and the second its value.

At present the only parameters which can be set *only* in the data file (aside from :action and the ":no-frequency" declaration) and not on the web input page are ipf-maxit and ipf-maxdev, which control the Iterative Proportional Fitting Algorithm, but the user will in general not need to think about these parameters or change them from their default values. IPF generates the calculated probabilities (q's) for a model. ipf-maxit is the maximum number of IPF iterations; ipf-maxdev is the maximum difference of frequencies (not probabilities) allowed between a state in the distribution for a calculated projection included in the model and the corresponding state in the observed projection. If Chi-square errors are reported in a run, consider increasing "ipf-maxit" and decreasing "ipf-maxdev."

One can in the data file specify the number of levels to be searched and the search width (the number of models retained at each level). For example, to search 10 levels and keep the best 5 models at each level, one adds the following lines above the data:

```
:search-levels
10

:optimize-search-width
5
```

However, one can specify the number of search levels and the search width on the web input page, and it is more convenient to do so there, because they are more easily changed. When search levels and width are specified *both* in the data file and on the web input page, the web input page values take priority. If these values are *not* specified in either the data file or the web input page, they will take on their default values, as follows:

Table 1. Data file parameters and their defaults

parameter	default
search-levels	12
optimize-search-width	20
ipf-maxit	266
ipf-maxdev	.25

Parameter specifications can be echoed in Occam's output by checking the "Print Options Settings" (see below) so that one can have a record of them. This is good practice, so echoing is *on* by default.

Comments in the data file

A line beginning with "#" will be ignored when Occam reads the data file, so this character can be used to begin comment lines. Also on any given line, Occam will not read past a "#" character, so comments can also be added at the end of lines which provide actual input to the program.

WEB INPUT

We now discuss the other parts of the web input page, as can be seen at

http://dmm.sysc.pdx.edu/occam/weboccam.cgi?action=search

GENERAL SETTINGS

Starting Model

Occam searches from a starting model. This can be specified on the browser page as "top" (the data or "saturated model"), "bottom" (the independence model), or some structure other than the top or bottom, e.g., "AB:BC". This field can also be omitted, in which case Occam uses the starting model specified in the data file (after the variable specification and before the data), as follows,

:short-model
AB:BC

("Short" refers to the variable abbreviations.) If the data file also does not specify a starting model, Occam uses the default starting model, which for neutral systems is "top", and for directed systems is "bottom".

Note that even though in Occam output of directed systems the component having all the IVs is abbreviated as "IV", such usage is not yet legitemate for specifying the starting model.

Composition Method

The default is standard, but one can also use the "Back Propagation (Fourier)" composition procedure to translate a model into a calculated (q) probability distribution. (This implements a mean square error minimization rather than an entropy maximization.) Once one has this distribution, the rest of the analysis – the calculation of transmission, information, %reduction of uncertainty, likelihood-ratio chi-square, and alpha – is standard. BP composition is not iterative and scales with the data and not the state space, so it is fast and can be done where the size of the state space could not be accomodated and thus IPF could not be done. However, the BP composition mode is experimental and is presently under investigation.

Reference Model

Assessing the quality of a model involves comparing it to a reference model, usually either "top" or "bottom". If the reference model specified in the browser page is left as default, it will be "top" for neutral systems and "bottom" for directed systems (like the convention for the starting model). If the reference model is "top," one is asking if it is reasonable to represent the data by a simpler model. If the reference model is "bottom," one is asking whether the data justifies a model more complex than the independence model.

The reference model can be the starting model. When the starting model is neither the top or the bottom, this can be used to determined whether "incremental" changes from the starting model are acceptable, as opposed to whether "cumulative" changes from the top or bottom are acceptable. The starting model may be a good model obtained in a prior search, and one may now be investigating whether it can be improved upon. At present, if the reference model is chosen to be the starting model, the starting model must be entered explicitly on the browser input page; Occam will not pick it up from the data file.

Models to Consider

Occam offers a choice between (a) all, (b) loopless, (c) disjoint, and (d) chain models.

All models

"All" means there are no restrictions on the type of model to be considered. One controls the extent of this search with parameters "search-levels" and "optimize-search-width", both of which must be set in the data file. Their current default values are 1000000 and 20, respectively. A width of 20 will generate the full lattice (all 114 models) for a 4-variable neutral system. Occam generates all "parents" of a model if search direction is "up" or all "children" if search direction is "down". It then retains the best "optimize-search-width" number of models, where best is determined by the parameter "During Search, Sort By", which is either Information or Alpha. (At the starting level, there is only one model, but subsequently there will always be "optimize-search-width" models.)

Loopless models

Loopless models are a subset of the full lattice of structures. For example, AB:BC is loopless, but AB:BC:AC has a loop, and would not be included in a loopless search. Doing a loopless search will be faster than an "all" search for two reasons: (1) the iterative procedure (Iterative Proportional Fitting, or IPF) used to generate model probabilities converges in a single cycle for loopless models, but requires several and possibly many cycles for models with loops, and (2) the lattice of loopless models is smaller than the full lattice.

An important use of a loopless search is for variable screening in directed systems. In a directed system, all models have one component which includes all the IVs, and all other components include at least one DV. Call a component that includes a DV a "predicting component"; these are shown in bold in this paragraph and the next. A *single-predicting-component* (SPC) model, e.g., AB:**AC**, will never have a loop, but *multiple-predicting-component* (MPC) models, e.g., AB:**AC**:**BC**, will always have loops. So a loopless search looks only at SPC models. This is valuable for screening IVs, i.e., for eliminating IVs which don't impact the DV(s) very much. Suppose one had 100 IVs and 1 DV, and wanted to find out which of the 100 IVs has predictive value for the DV. A loopless search will provide this information.

For a loopless search, "search-levels" determines how many IVs will be in the SPC, and "optimize-search-width" determines whether all such models are considered. To illustrate: suppose, one has four IVs, A,B,C,D, and one DV, Z, and one starts the search at the bottom. If "optimize-search-width" is 2 and "search-levels" is 3, then at the first search level Occam generates all parents of ABCD:**Z**, i.e., all one-IV SPC models: ABCD:**AZ**, ABCD:**BZ**, ABCD:**CZ**, ABCD:**DZ**. On the basis of the Sort parameter specified in the browser input page, Occam then picks the best 2 of these, say ABCD:**BZ** and ABCD:**DZ**. Then, at the second search level, all parents of these 2 models are considered. These will include predicting components of **ABZ**, **CBZ**, **DBZ**. and **ADZ**, **BDZ**, **CDZ**. The best 2of these 5 models will be retained. Say these are ABCD:**ABZ** and ABCD:**BDZ**. Occam then examines, at the third search level, all parents of these models, and again keeps the best 2.

If one wants to do an *exhaustive* search of *all* SPC models with a certain number of IVs in the predicting component, one needs to set the width parameter high enough. For problems with many variables, if the number of IV predictors one wants to consider is high, this may be impractical. A *heuristic* selection of good SPC models may then have to be done, using reasonable values of "optimize-search-width" and "search-levels".

Disjoint models

"Disjoint" means non-overlapping, that is, any two components of a model do not overlap in their variables. For neutral systems, the idea of a disjoint model is straightforward. A disjoint model search would reveal what are the best "cuts" of a system into non-overlapping subsystems, e.g., for a 4-variable system, AB:CD or AC:B:D. Such a search could also be used as a rough search, after which one might do a downwards search relaxing the constraint of disjointness.

For directed systems, the notion of a disjoint model is not as straightforward. Only the independence model and the saturated model are disjoint in a strict sense. For example, for three variables where A and B are IVs and C is the DV, since every directed system model must have an AB component, only AB:C and ABC are disjoint. What one is really interested in here is the disjointness of the *predicting components*, and more specifically, the disjointness of the *IVs in the predicting components*. A disjoint model, for a directed system, will thus be defined to mean that there is no overlap in the IVs of any two predicting components. That is, the influence of subsets of the IVs on the DV are separable, and have no interaction effects. For example, directed system ABC:**A**Z:**B**Z is disjoint, but directed system ABC:**AB**Z:**BC**Z is not. Note that if ABC:AZ:BZ were a neutral system, it would be considered *not* disjoint.

In summary, for neutral systems, disjoint models partition all the variables into non-overlapping subsets. For directed systems (with one DV), disjoint models partition all the IVs which affect the DV into non-overlapping subsets.

Chain models

AB:BC:CD:DE illustrates the idea of a chain model. All components have two variables, and every component, except for the ends, overlaps the component to the left with one variable and the component to the right with the other. Chain models searches are not searches in the sense of starting with a model and going either up or down the lattice. Occam simply generates and evaluates all chain models. Chain are currently being used for studies on the use of RA to prestructure genetic algorithm genomes. One could compare all possible lineal causal chains, of the form A → B → C → D, by using the chain model option.

Search Direction

The default direction is up for directed systems and down for neutral systems, but for some purposes one might wish to do a downwards search for a directed system or an upwards search for a neutral system. The Search Direction should not be confused with the Reference Model. Model assessments depend on the Reference Model but not on the Search Direction.

During Search, Sort By

The browser page offers a choice of sorting by Information, or Alpha. The best "optimize-search-width" models at every level which are retained for going to the next level are determined by this sort criterion.

Information is constraint captured in a model, normalized to a range of 0 to 1. It is linear with uncertainty (Shannon entropy), likelihood-ratio Chi-square, and %-reduction of uncertainty (for directed systems with one DV), so sorting on information is *equivalent* to sorting on one of these parameters.

Alpha is obtained from Chi-square tables using the likelihood-ratio Chi-square and dDF (delta-degrees of freedom) as inputs. It is the probability of a Type I error, namely the

probability of being in error if one rejects the null hypothesis that a model is really the same as the reference model. Note that if the reference model is "bottom", a model is good, in the sense of being statistically different from the independence model, if Alpha is *low*, so the "standard" cut-off of 0.05 could be used. If, the reference model is "top", a model is good, in the sense of being statistically the same as the data, if Alpha is *high*, so the standard 0.05 makes no sense. However, we don't want Alpha to be too high, or the model will be too complex. In one log-linear book, an Alpha of .1 to .35 is recommended, but the choice of Alpha really depends on the user's purposes.

When Searching, Prefer

At every level Occam chooses the best "width" out of a set of candidate models by using the sorting criterion (Information or Alpha). When this criterion is Information, one obviously prefers Larger Values, but when the sort criterion is Alpha, one might prefer *either* "Larger Values" (if the reference model is the top and one cares a great deal about fidelity to the data) or "Smaller Values" (if the reference model is the bottom and one cares a great deal about the statistical justifiability of complex models.

Search Width

This is the number of the best models retained at every level. If the value is specified it overrides any value specified in the data file. If the value is omitted, the value in the data file is used, and data file also does not specify a value, the default value of 20 is used. (This value will cause *all* models for a four-variable neutral system to be examined.)

Search Levels

This is the number of levels to be searched, *including* the starting model. If the value is specified it overrides any value specified in the data file. If the value is omitted, the value in the data file is used, and if the data file also does not specify a value, the default of 12 is used. (This value will cause *all* models for a four-variable neutral system to be examined.)

REPORT SETTINGS

In Report, Sort By

Output can be sorted by (a) Information, (b) Alpha, (c) dDF, and (d) levels. (NB: the measure used to sort the Occam output report need not be the same as the measure used in sorting done in the search process.) dDF is the change of degrees of freedom relative to the reference model. Sort by levels allows the user to have output which truly follows the order of the Lattice of Structures; this is not actually accomplished by sorting on dDF, because different variable cardinalities can result in a model at a lower level still having a higher DF than a model at a higher level.

In Report, Sort

Occam output can be printed in either (a) descending or (b) ascending order of the magnitudes of the sorting measure. For example, if the report is sorted on Information in a descending order, then the most complex, high information, models will appear in the output at the top of the page.

Add to Report: BP-based Transmission

If checked, Occam will add BP-based Transmission to the measures normally outputted for standard composition. This allows the systematic study of the similarities and differences between standard and BP-based composition. BP-based composition and the BP transmission are advanced experimental features of OCCAM under current investigation.

Add to Report: Percent Correct

If checked, Occam will add Percent Correct to the measures normally outputted. This is a measure of goodness of a model very different from information or amount of uncertainty reduced. It is relevant where one wishes to predict from the values of the independent variables what the value will be for a dependent variable. Percent correct is defined as
$(1/N) \sum_k N(k, jmax(k))$, where N is the sample size, k is an index which runs over IV states, j is an index which runs over DV states, N(k,j) is the number of cases having IV_k and DV_j, jmax is the j which gives the highest calculated probability, $q(DV_j \mid IV_k)$, for the model under consideration. To read about the use of Percent Correct, see:
http://www.sysc.pdx.edu/download/papers/heartIJCNNabstract.htm

Return Data in Spreadsheet Format

If this is selected, Occam returns its output as a .csv (comma separated columns) file, where the first name of the file is the first name of the input file. The .csv format is one of the standard input formats for Excel, so if one clicks on the .csv file, one will go directly into Excel, and see the Occam output in an Excel spreadsheet for further processing. While Occam seems to allow the user to either open or save the .csv file, in actual fact, the user must save the file and open it later.

Print Option Settings

When selected (this is the default), Occam echoes the parameter setting which have been specified in both the browser input page and the data file before it displays the actual output of the Occam run. This allows the user to document what data file and parameter settings produced the Occam output.

Run in Background, Email Results To:

For jobs that are likely to take too long to wait for immediate browser output, type in your email address, and Occam will email the results to you in spreadsheet format.

You can check the status of your job by choosing **Show Log** on the main Occam page and typing in your email address. The log contains *two lines* for every job submitted for background running. When the job is submitted, the log adds the line "*Job started: data/ filename*." When the results are emailed to the user, a second line is added: "*Results for data/filename sent to username@email_address*."

Send

This sends the browser page to the Occam server. Occam will return its output in a new window. This makes it easy for the user to change parameter settings on the browser input page, and resubmit.

When jobs are submitted to run in background, the browser will first say: "*Batch job started*:" When the data file has been read in, and the background job has been started, the browser will add: "*data file: filename, received from* usernameatemailaddress". Do not close this 2nd browser window until after you see this 2nd line appear.

IV. SEARCH OUTPUT

If Print Options Settings has been selected, the Occam output will begin by echoing the parameter settings from the web input page and from the data file. At present Occam also outputs the values of "search-levels" and "search-width" even if these have not been explicitly specified in the data file (the latter is specified as "optimize-search-width"); this tells the user what the default values currently are.

Occam will always print out as it proceeds from level to level how many models are generated at each level and how many of these are kept. This lets the user track the progress of Occam. It also shows whether an exhaustive search is being done (all models generated are kept) or only a partial (heuristic) search is being done (only some generated models are kept, i.e., the lattice is being pruned).

OUTPUT FILE FOR DIRECTED SYSTEM

Below are the output for the data given as an example in the DATA FILE section. This is a directed system with the DV being C and the IVs being A and B. The output has been sorted on Information.

MODEL	Level	H	dDF	LR	Alpha	Information	%dH(DV)
ABC	3	2.76118	3	10.61218	0.013975	1	0.563878
IV:AC:BC	2	2.761567	2	9.820459	0.007371	0.925395	0.52181
IV:BC	1	2.761822	1	9.297822	0.00207	0.876147	0.49404
IV:AC	1	2.766346	1	0.028393	0.866041	0.002676	0.001509
IV:C	0	2.766359	0	0	1	0	0

In the Model field, "IV" means a component with all the IVs in it, here AB. H is information-theoretic uncertainty (Shannon entropy). dDF is delta-degrees of freedom, the difference in df between a model and the reference model. The model for which dDF is 0 is the reference model. LR is the Likelihood-Ratio Chi-square (L^2 in Krippendorff), which is the error between a model and the reference model. Alpha is the probability of a Type I error, namely the probability of being in error if one asserts that the model is really the same as the reference model. Information is a measure of the constraint captured in a model, normalized to [0,1] range, namely [T(bottom) – T(model)] / T(bottom), where T is transmission. %dH(DV) is the % reduction in uncertainty of the DV (if there is only one DV) given the IVs in the predicting components (note that for this data, the reduction of uncertainty is very small, less than 1% even if one predicts

with both IVs and these IVs interacting. Information is a standardized measure, scaled from 0 to 1, but %dH(DV) is the actual reduction of uncertainty achieved by any model. %dH(DV) exactly equals Information multiplied by the %dH(DV) for the top (saturated) model. For more information on these measures, see the "Wholes and Parts" and "Overview of Reconstructability Analysis" papers mentioned above.

Note that only dDF, LR, and Alpha depend on the choice of reference model. Values of H, Information, and %dH(DV) are "absolute" and do not depend on reference model.

OUTPUT FILE FOR NEUTRAL SYSTEM

If C is regarded, along with A and B, as an IV, then the system is neutral. Below are the measures for the larger lattice of neutral systems. Note that the column for uncertainty reduction is omitted because there are no DVs.

MODEL	Level	H	dDF	LR	Alpha	Information
ABC	0	2.76118	0	0	1	1
AB:AC:BC	1	2.761567	1	0.791718	0.373511	0.987028
AB:BC	2	2.761822	2	1.314355	0.518312	0.978465
AB:AC	2	2.766346	2	10.58378	0.005032	0.826589
AB:C	3	2.766359	3	10.61218	0.013975	0.826123
AC:BC	2	2.786416	2	51.70655	0	0.152807
A:BC	3	2.78643	3	51.73495	0	0.152341
AC:B	3	2.790954	3	61.00438	0	0.000465
A:B:C	4	2.790968	4	61.03277	0	0

ERROR AND WARNING MESSAGES FROM OCCAM

The following error and warning messages may appear in the search output.

1) Cardinality Error:

If the user specifies a value of Cardinality less than the total number of states present in the data for the variable, a warning will be issued -
"Too many different values for variable x"
Occam will ignore any states more than specified cardinality.

In case of Cardinality being greater than number of states, the following warning will be issued

Input data cardinality error variable x specified cardinality 4 <> 3

The analysis presented by Occam in such situations may not be valid and therefore care should be taken to make sure the Cardinality of the variable is correct.

2) Start and reference Model Errors:

If the model specified as Start or Reference Model in the data file or in the web menu happens to be an Invalid model (e.g. IV:AD:BD) , Occam will issue an error message and will terminate.
"ERROR 99

Occam error: invalid model name"

3) Rebin string errors

If the rebinning string is incorrectly formed, occam will issue an error and will terminate.
It will be a 200 level error.
"Error 2xx
Error in Rebinning string"

4) No data specified error

If the ":data" tag is missing or there is no data following the tag, Occam will report an error, stating no data was found.

5) Rebinning an ignored variable warning

If the variable is marked to be ignored and the rebinning string is present. In this case Occam will ignore the rebinning string and the analysis will be done without rebinning. Occam will issue a small warning:"For variable =>x rebinning parameters will not be considered since it is marked for no use."

V. FIT INPUT

The Fit option is designed to give the user a more detailed look at a particular model. That is, Search examines many models and outputs different measures to characterize these models. Fit similarly outputs many measures for a particular model, but more critically it outputs also *the actual model* itself, not just its name. That is, it outputs the calculated frequency/probability distribution for the model.

Fit takes the same input file described above for Search. The web input page is, however, much simpler as can be seen at

http://dmm.sysc.pdx.edu/occam/weboccam.cgi?action=fit

Only the data file name/location, and the model to be fit need to be specified. (In addition, the output can be specified to be in spreadsheet format, and Occam can be directed to email it to the user.)

VI. FIT OUTPUT

After echoing the input parameters (which are requested by default), Occam prints out some properties of the model and some measures for the model where the reference model is first the top and then the bottom of the lattice. For now, values printed out for *beta*, the probability of a Type II error, should be ignored, as their correctness has not yet been checked.

After this, Occam prints out for every cell, the observed and calculated probability and the difference between the two (the residual).

NOT YET IMPLEMENTED (available now in the older program Occam2, not yet available in Occam3): After this, for directed systems, Occam outputs for every IV state the *conditional* frequencies of the different DV states, expressed as %'s. In addition, above this table it prints out the marginal frequencies of the DV states, also expressed as %'s. This allows the user to see by inspection for which IV states are the conditional frequencies noticeably different from the marginal frequencies.

VII. SHOW LOG

This lets the user input his/her email address and see the history of the batch jobs which have been submitted and the Occam outputs for these jobs which have been emailed back to the user. The web page looks as follows:

http://dmm.sysc.pdx.edu/occam/weboccam.cgi?action=showlog

VIII. FREQUENTLY ASKED QUESTIONS

1. How do I determine the best predictor or best set of IV predictors of some dependent variable?

Do an upwards search, from the independence (bottom) model, IV:DV, using this also as the reference model, looking only at loopless models.

If you going to use a "saturated" model, i.e., with all the IVs in one predicting component, then stop this upwards search at the point where adding IVs is not statistically significant. *But* if you are willing to use a multi-predicting component model (the subject of question #2), then you can, in this upwards search, add IVs *beyond the point* that the model is statistically significant (i.e., beyond thepoint where alpha is very small), since you will next (as the answer to question #2 indicates) be doing a *downwards* search towards models of lower complexity. In this second search, you may obtain a statistically significant multi-component model using *all* the IVs you found in the first search – but in components each having only a subset of them.

To illustrate: say you are prepared to accept a model only if alpha (probability ot a Type I error) is equal or less than .05. Suppose that the best model which satisfies this, i.e., the most complex model which is statistically justified, is IV:ABCZ, which, say, reduces the uncertainty of Z by 10%. In the first search, you might go beyond this model up to model IV:ABCDZ, which reduces the uncertainty by 15%, but has alpha = .1. In the 2nd (downward) search, you might then arrive at IV:ABZ:BCZ:CDZ, which reduces the uncertainty by 12%, i.e., is better than IV:ABCZ, and has alpha = .04. Note that the model IV:ABZ:BCZ:CDZ uses all of the IVs in model IV:ABCDZ, for predicting Z, but

in *smaller subsets*. IV:ABZ:BCZ:CDZ thus has lower df than IV:ABCDZ, and thus can be statistically significant, while IV:ABCDZ is not.

If you are interested only in the best single IV predictor, you need only to do this upwards search for one level. If you want to see several IVs ranked by their predictive power, set "optimize-search-width" in the data file to the number of single predictors you want reported. For example, if it is set to three, what will be reported is the best single predictor, the 2nd best single predictor, and the 3rd best single predictor. If you want the best *pair* predictors, go *two* levels up; again the width parameter will indicate how many of these will be reported.

2. How do I determine the best multi-predicting component model for some set of IV predictors?

Do a downwards search from the saturated model containing all the IVs, using the independence model as the reference, look at all models (i.e., models with loops).

At the present time, the number of IVs for such searches should not exceed 10, and in the 7-10 range, the search may take a while, depending on what the search width is.

3. For what purposes are loopless models used for directed systems?

Loopless models, for directed systems are models which have a single predicting component, in addition to a component defined by all the IVs. Loopless models are used to find a best set of IV predictors; see question #1.

4. For what purposes are disjoint models used for directed systems?

Disjoint models are models with loops, but do not have any IVs which occur in more than one predicting component. For example, ABCD:ABZ:CDZ is a disjoint directed system model, while ABCD:ABCZ:CDZ is not, since C occurs in two predicting components . Using disjoint models instead of all models can speed the search. It also partitions the IVs into separate groups, which may make model interpretation simpler.

Each grouping of IVs (the IVs in each component) might perhaps be thought of as defining a latent variable.

5. How do I know if there is an interaction effect between IVs in predicting a DV?

For simplicity consider two predicting IVs, A and B, from a larger set of IVs. Start an upwards search with a disjoint model where each IV predicts the DV separately, i.e., ABC…:AZ:BZ. Use this model not only as the starting model but also as a reference model. (In the Occam input page, for Reference Model, select the choice which sets it as the same as the Starting Model.) In the upwards search the alpha for ABC…:ABZ indicates there is an interaction effect if its value is acceptably low (statistically significant) and if it reduces the uncertainty of Z by more than the reference model.

Suppose now one has three IVs, namely A, B, and C. If one tests whether ABCD…:ABCZ is statistically significant relative to a reference model of ABCD…:AZ:BZ:CZ, one will ascertain whether some interaction effect is present, but if one wants to be sure that this interaction effect involves all three variables, then one should start the search and use as a reference model ABCD…:ABZ:ACZ:BCZ. Then if the transition between this model and ABCD…:ABCZ is statistically significant, one knows that there actually is an interaction effect involving all three IVs.

6. How many bins shall I bin my quantitative variables into?

Binning can be done "rationally," i.e., using substantive knowledge about how qualitatively distinct values ought sensibly to be defined, or "technically" by some mathematical procedure, without regard to substantive issues of interpretation. For example, plotting your data on a histogram and assigning bins to clear and natural groups is a rational procedure, but be aware that if these groupings put very many cases into one bin and only a few into others, one is losing discriminating power by such a binning assignment.

For binning technically, 3 bins is a good default, since it allows for a single variable the detection of a non-linear relation, while 2 bins does not. More bins will give finer discrimination but bins should be thought of as a resource to be optimally distributed among all the variables. The total number of bins, i.e., the product of the number of bins for all variables, should, by conventional wisdom, be about a fifth of the sample size, or to put it the other way, the sample size should be 5 times the number of bins (the size of the state space). In practice, setting the number of bins equal to the sample size often works, but although one might be able to decide on good vs. bad models with smaller sample sizes relative to the state space, one is much less likely to be able to make reliable statements about particular states.

7. When should I do an upwards search and when should I do an downwards search?

The Occam default is an upwards search for directed systems and a downwards search for neutral systems, but you could, if you wanted to, do the opposite (a downwards search in a directed system or an upwards search in a neutral one). As a general rule, do an upwards search when the reference model is the bottom (the independence model). In this case, you are interested in ascending the lattice as high as you can – for directed systems, in gaining maximum predictive power – as long as the complexity of the model is statistically justified. Similarly, as a general rule, do a downwards search when the reference model is the top (the data). In this case, you are interested in getting as low as you can – in finding the simplest model that satisfactorily fits the data.

8. Why is it necessary to set the number of levels to be searched and the search width in the data file, as opposed to the web input page, where most other parameters are set?

We just haven't yet gotten to putting these parameters on the web input page.

9. Why are models with high alpha better for downwards searches, and how high should alpha be?

In downwards searches, the null hypothesis is usually that a model is the same as (agrees with) the data. The probability of a Type I error means the probability of being wrong in rejecting this hypothesis that the model agrees with the data. For a model we are hoping to accept, we want alpha to be high because we want to be sure that we would be wrong if we said that the model differs from the data.

How high alpha should be is a user choice, and depends also on how important it is to the user that the model obtained be relatively simple. The point is that it should definitely greater than the .05 that one might use rationally for upwards searches. If one had a model with alpha = .05, where the reference was the top and not the bottom, one would be selecting a model that one is virtually certain is different from the data, clearly an irrational choice. The Sage log-linear book suggests that one might therefor increase alpha to about .3, but this is completely arbitrary; one could just as well want alpha to be .7 or .8.

10. In a spreadsheet I found that for directed systems %reduction in DV uncertainty and %information are proportional to one another. Why does Occam bother to print them both, if they are so simply related?

Just to save the user from having to do some extra computing. %Information is just equal to %uncertainty reduction of a model divided by the %uncertainty reduction of the top (saturated) model.

%Information is standardized to a 0-100% range, and indicates how well any model compares to the top model. %reduction in uncertainty gives the actual numbers of uncertainty reduction for all models; the top model might reduce uncertainty a lot or a little.

11. What is the fit option and how is it different from the search option?

One uses search to find a good model or set of models. One uses fit to look at a model in greater detail.

12. Fit doesn't seem to do very much; how come?

Only a few of the features of the old Occam (Occam2) for fit have been so far implemented in the new Occam (Occam3). More will "soon" (hopefully) be added.

13. How would I test the hypothesis that B "mediates" an effect of A on the DV, Z?

This hypothesis means a causal model, A → B → Z. In RA terminology, this means model AB:BZ. To test the hypothesis that this is a good model, one tests the statistical significance of the difference between this model and the data. That is, one has the reference model being the top, and one wants the AB:BZ model to have high information and also high alpha.

Technically, one here would like to know the value of beta, the probability of making an error in accepting (not in rejecting) the hypothesis that AB:BZ is the same as the data. One would like this beta to be low. Unfortunately, Occam3 right now does not offer any calculation of beta (it may in the future), and one has to make do with its calculation of alpha, which one wants to be high. (In general there is a tradeoff between alpha and beta, so that when alpha is high, beta is low, but beta is NOT simply 1-alpha.)

Note that the model AB:BZ does not actually require the above causal interpretation. It could also be interpreted as A → B ← Z or A ← B ← Z. That is, RA does not and cannot distinguish between these situations, and an argument that it is one rather than another has to be made by the user.

14. I am doing a downwards search with the top as my reference model, and I find that any decomposition results in a severe drop in alpha. Does that mean that I cannot decompose the data at all?

Not necessarily. This effect could be due to your having a very large sample size (at least relative to the state space), so that any deviation from the data is statistically significant. In such situations, you can just not make decisions based on statistical significance and instead make decisions based on %Information. That is, you can go down the lattice of structures as far down as you can as long as %Information is greater than some minimal value of your choosing.

15. What are chain models and how are they useful?

Chain models for directed systems are models like IV:ABZ:BCZ:CDZ and for neutral systems are models like AB:BC:CD. At present these models are being used in a project using RA as a preprocessor for genetic algorithms. They may or may not be of more general usefulness.

16. Of the search outputs, what measures depend on the reference model, and what measures do not?

LR (likelihood ratio, which is the same as L^2) and alpha depend on the reference model which is chosen for the Occam run. Entropy (uncertainty), %Information, and %Uncertainty reduced do not depend on the reference model, that is, they are inherent properties of each model regardless of the reference model chosen for the run. Level and dDF depend upon the reference model (which by definition has Level = 0 and dDF = 0). Level does not depend on the actual data, i.e., is purely about the structures of models

and not about their distributions. dDF depends on the data only in its dependence on the cardinalities of the variables; it does not depend on the actual observed distribution at all.

17. Of what value is the printout of numbers of models generated and kept which gets printed before the actual search output?

By looking at the numbers of models generated and kept at each level, and at the running totals for these numbers, you can get a sense of how much the width parameter is pruning the search tree, i.e., how many models are being discarded as you go from one level to the next.

The width parameter is set at a default of 20, so that for a four variable neutral system, search will generate and keep *all* models in the lattice, that is, you will be doing an *exhaustive* search. For more variables, you would have to increase width to do an exahustive search, and this rapidly becomes impractical, so that one has to do a search which only samples the lattice.

18. Loopless searches seem to be pretty fast, but searching all models often takes very long. How come, and is there some way to speed up all-model searches?

Loopless searches don't need IPF, and go with the data and not the state space.

At present all-model searches need IPF and go with the state space and not the data, so these searches will necessarily take a long time. The Fourier composition approach may allow all-model searches to be done as fast as loopless searches, but this is still experimental at this point.

19. What about set-theoretic RA?

Not yet implemented in Occam3. Available in a separate program.

20. What about latent variable models?

Not yet implemented in Occam3 or in any separate RA program. However, latent variable log linear programs exist (though they work in the confirmatory, not the exploratory, mode, so they do not search many models).

21. What about state-based models?

An experimental Excel program is being developed by Michael Johnson. Work is beginnng on incorporating SB modeling into Occam.