Dipartimento di Statistica "Giuseppe Parenti"

Dipartimento di Statistica "G. Parenti" – Viale Morgagni 59 – 50134 Firenze - www.ds.unifi.it

WORKING PAPER 2004/12

A Comparison of Complementary Automatic Modeling Methods: RETINA and PcGets.

Teodosio Perez-Amaral,
Giampiero M. Gallo, Hal White

Università degli Studi
di Firenze

Econometrics. Statistics

This version: Oct. 4, 2004

A Comparison of Complementary Automatic Modeling Methods: RETINA and PcGets.

TEODOSIO PEREZ-AMARAL[†], GIAMPIERO M. GALLO[‡] and HAL WHITE[*]

[†] *Departamento de Análisis Económico, Universidad Complutense de Madrid 28223 Madrid, Spain*
(e-mail teodosio@ccee.ucm.es)

[‡] *Dipartimento di Statistica "G. Parenti", Università di Firenze, Viale G.B. Morgagni, 59 - 50134 Firenze, Italy*
(e-mail gallog@ds.unifi.it)

[*] *Department of Economics, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA, 92093-0508, USA*
(e-mail hwhite@weber.ucsd.edu)

ABSTRACT

In Perez-Amaral, Gallo, and White (2003), the authors proposed an automatic predictive modelling tool called Relevant Transformation of the Inputs Network Approach (RETINA). It is designed to embody flexibility (using nonlinear transformations of the predictors of interest), selective search within the range of possible models, control of collinearity, out-of-sample forecasting ability, and computational simplicity. In this paper we compare the characteristics of RETINA with PcGets, a well-known automatic modeling method proposed by David Hendry. We point out similarities, differences, and complementarities of the two methods. In an example using US telecommunications demand data we find that RETINA can improve both in- and out-of-sample over the usual linear regression model, and over some models suggested by PcGets. Thus, both methods are useful components of the modern applied econometrician's automated modelling tool chest.

Note: We thank Peter C.B. Phillips and two anonymous referees for constructive comments and for pointing out directions for further research. The first and second author acknowledge financial support from the Coordinated Exchange programs funded by the Italian and Spanish Ministries for University and Research. Thanks are due to Massimiliano Marinucci, Universidad Complutense, for performing the computations of the empirical example.

1. Introduction

Model building and specification selection is a process blending science and art: finding candidate models must adhere to some sort of paradigm, as Ploberger and Phillips, 2003, make clear in their discussion of the relationship between the data generating process and a principle of parsimony in achieving an adequate representation of the data. Recent developments have gone in the direction of making some of this process automatic, and hence implementable in software. A leading method is the strategy known as general-to-specific (Gets) modeling proposed by David Hendry, which Hoover and Perez (1999) first suggested to implement in an automated way. An overview of the literature, and the developments leading to Gets modeling in particular, is provided by Campos, Ericsson and Hendry (2004). Finite-sample behavior is examined in Krolzig and Hendry (2001) and Hendry and Krolzig (2003). PcGets, a computer program that implements the Gets modelling is described in Hendry and Krolzig (2004).

PcGets has four basic stages in its approach to selecting a parsimonious undominated representation of an overly general initial model, denoted the general unrestricted model (GUM). The first concerns the estimation and testing of the GUM; the second is the pre-search process; the third is the multipath search procedure; and the fourth is the post-search evaluation. See Hendry and Krolzig (2001) for details.

In this paper we compare PcGets with RETINA, another automatic modeling method, recently proposed in Pérez-Amaral, Gallo and White (2003), based on earlier work by White (1998). It aims at achieving a flexible and parsimonious prediction model that well approximates the conditional mean of a variable, given a (potentially large) set of variables of interest, in situations where one does not have specific information as to the functional form of the conditional mean or as to the relevance of individual variables. RETINA has the flexibility of neural network models (see White, 1989) in that it accommodates nonlinearities and interaction effects (through nonlinear transformations of the potentially useful variables in the conditioning set), the concavity of the likelihood in the parameters of the usual linear models (which avoids numerical complexity in estimation), and the ability to identify a set of

attributes that are likely to be truly predictive (which corresponds to a principle of parsimony). In performing model selection, the approach relies on an estimation/cross-validation scheme that is aimed at limiting the possibilities that good performance is due to sheer luck.

In section 2 we outline the basic features of RETINA. In section 3 we compare and contrast the objectives, strategies, selection criteria, base models, and other features of RETINA with those of PcGets. In section 4 we illustrate the complementarity of the methods by applying both to real data on US telecommunications demand. Section 5 contains concluding remarks.

2. RETINA.

The rationale behind the RETINA procedure is a predictive modeling effort guided by the data structure. In general there is no theory required apart from some guidance in selecting an information set of potential predictive usefulness. The key principles that have inspired the construction of the algorithm are flexibility, selective search, control of collinearity and out-of-sample predictive ability. The procedure is described in detail in Table 1 below (cf. also the original paper by Perez-Amaral, Gallo and White, 2003); here it is helpful to sketch the motivations for these basic features of RETINA.

Flexibility: Flexibility is required to handle the lack of information about the functional form of the conditional mean of the dependent variable *Y* given "inputs" *X* that is common in economics. To attain this flexibility, one may use a set of transformations of the input variables, say $\zeta(X)$ = { $\zeta_j(X)$, j = 1, …, m}, that embodies both *nonlinearities* and *interactions*.

Selective Search: The task of evaluating all 2^m possible models arising when we have m candidate regressors in the set of transformed variables $\zeta(X)$, and then of applying some form of model selection becomes impossible for even a moderate value of m. Rather, following the ideas in White (1998), we can select from a judiciously chosen subset of all possible models (of order proportional to m), admitting (transformed) predictor variables on the basis of their relevance for the problem at hand. For example, one may rank the candidate regressors according to their correlation in absolute value with the dependent variable, and include them in the model sequentially in rank order.

Collinearity: In order to control the degree of dependency among the variables of the model, one can require that the amount of collinearity between the included predictors and any candidate predictor considered for sequential inclusion must lie below a threshold parameter λ chosen by the experimenter (as λ approaches 0 new regressors approach orthogonality; as λ approaches 1 new regressors may be highly collinear).

Out-of-Sample Forecasting Ability: Although flexibility is desirable, it creates the danger of overfitting the sample data. In order to avoid this, we use disjoint sub-samples for estimation and cross-validation and an out-of-sample prediction performance criterion for model selection as important features of the procedure.

TABLE 1 - The RETINA algorithm†

Stage 0 - Preliminaries

1. **Data building and sorting**
   a. Generate the set of transformed variables $\zeta(X) = \{W_1, \ldots, W_m\}$.
   b. Randomly divide the sample into three sub-samples.

Stage I - Isolating a "candidate" model

2. **Using Data on the First Sub-sample**
   a. Order the variables in ζ(X) according to their (absolute) sample correlation with the dependent variable in the first sub-sample alone. Let $W_{(1)}$ be the variable with the largest absolute correlation with Y, $W_{(2)}$ be the second most correlated, and so on.
   b. Consider various sets of regressors all of which include a constant and $W_{(1)}$: each set of regressors $\zeta_\lambda(X)$ is indexed by a "collinearity threshold" $0 \le \lambda \le 1$ and is built by including $W_{(j)}$ (j=2,…,m) in $\zeta_\lambda(X)$ if the R^2 of the regression of $W_{(j)}$ on the variables already included is less than or equal to λ.
   c. The number of sets of regressors is controlled by the number of values of λ between 0 and 1 chosen, say, ν.
3. **Using Data on the First and Second Sub-sample**
   a. Estimate each model by regressing Y on each set of regressors $\zeta_\lambda(X)$ using the data on the first sub-sample only and compute an out-of-sample prediction criterion (the cross-validated mean squared prediction error) using the data on the second sub-sample only. This involves the estimation of ν models.
   b. Select a "candidate" model as the one corresponding to the best out-of-sample performance $\zeta_{\lambda^*}(X)$.

Stage II – Search Strategy

4. **Using Data from both the Second and Third Sub-sample**
   a. Search for a more parsimonious model: estimate all models including a constant and all the regressors in $\zeta_{\lambda^*}(X)$ one at a

time in the order they were originally included, but also in the order produced by the procedure sub 2.a, this time on the basis of the correlations in the second sub-sample.

b. Perform an evaluation of the models out-of-sample (using the data on the third sub-sample) calculating a performance measure (the cross-validated mean squared prediction error, possibly augmented by a penalty term for the number of parameters in the model).

Stage III – Model Selection

5. **Repeat Stage I and II Changing the Order of the Sub-samples; Produce a Candidate for Each Sub-sample Ordering**
6. **Select the Model which has the Best Performance over the Whole Sample**

† Codes for running RETINA (GAUSS, MATLAB) are available upon request.

Some comments on the RETINA algorithm are in order.

- *Division of the entire sample into three sub-samples.* This is done in order to cross-validate using truly out of-sample-data.
- *Safeguards against spurious correlations.* This is achieved by three features of RETINA. One is the scrambling of the regressors by their correlation in the second subsample, another is the use of the parameter λ to control for collinearity, and a third is the repetition of the whole procedure on the different orderings of the sample.
- *Information criteria.* In order to select the final model associated with a specific ordering of the sub-samples we apply an information criterion; RETINA uses the out-of-sample AIC.

3. PcGets vs RETINA.

In this section we sketch similarities and differences between PcGets and RETINA. For clarity and brevity, we omit details that can be found in the references. We start by comparing the objectives, strategies and general characteristics of the two procedures, continue with some more specific details, and finally outline certain parallel aspects of the two methods.

A. Goals

PcGets	RETINA
a. Select a parsimonious undominated representation of an overly general initial model, the general unrestricted model (GUM) b. Best model fit within sample. c. Congruent with theory.	Identify a parsimonious set of (transformed) attributes likely to be relevant for predicting out-of-sample.

In many practical applications, the general-to-specific approach often proceeds as if the DGP is nested within the GUM (or, equivalently, as if the GUM is an overparameterization of the DGP), so that the DGP can be discovered by eliminating the irrelevant variables (this is not an explicit assumption of PcGets, though).

RETINA is less ambitious. It tries to approximate an unknown predictive relationship using a model that is not necessarily correctly specified, using transforms of the attributes to the extent that they contribute to out-of-sample prediction and are not too collinear with each other.

B. Strategy

PcGets	RETINA
a. General to specific. b. Formulate a general unrestricted model, GUM, and reduce it to a parsimonious model using residual tests and hypothesis testing on coefficients.	a. Specific to general: Start from a model with a single transform. Add additional transforms only if they contribute to out-of-sample forecast ability. b. Flexible and parsimonious model. c. Selective search of transforms. d. Control for collinearity.

The modeling strategies are quite different: PcGets proceeds from general to specific; it starts from a general model and reduces it down. RETINA starts from a model with just a constant and augments it by including additional transforms only when they help forecast out-of-sample and are not too collinear with previously included variables (no specific hypothesis testing is performed).

C. Base Model

PcGets	RETINA
GUM: General Unrestricted Model, specified by the researcher, usually based on theory. Using transforms of the original variables.	Based on original inputs and transforms, automatically selected from the first subsample by cross validation in the second, controlling for collinearity.

The role of the GUM is crucial in PcGets, but there is no automatic procedure for specifying the GUM, although there are useful guidelines. On the other hand, RETINA constructs several base models (GUMs in PcGets parlance) in an automatic fashion and then uses a strategy to reduce them using out-of-sample performance criteria.

D. Flexibility

PcGets	RETINA
The GUM determines maximum flexibility. May include transforms of the original variables.	a. The permitted transformations of the inputs determine maximum flexibility. b. The actual flexibility of the candidate model is chosen by the procedure

E. Selective/systematic search.

PcGets	RETINA
a. Starting from the GUM, performs a systematic search using multiple reduction paths. b. Using diagnostics, checks the validity of each reduction until terminal selection. c. When all paths are explored, repeatedly tests models against their union until a unique final model is obtained.	a. Uses a selective search to avoid the heavy task of evaluating all 2^m possible models and of applying some form of model selection. b. A saliency feature of the transforms, such as the correlation with the dependent variable, is used to construct a natural order of the transforms in which they are considered. c. Only a number of candidate models of order proportional to *m* is considered.

F. Collinearity.

PcGets	RETINA
a. Seeks to formulate the GUM, in search for a relatively orthogonal specification. b. A quick modeller option is available in PcGets for non-expert users.	Controls for collinearity by adding an additional transform to the candidate list only if the collinearity is below a certain (user defined) threshold.

The control for collinearity on the variables as they are expressed in the initial information set and subsequent transformations (hence nonsingular linear transformations are excluded from consideration as candidate regressors) is a priority within the RETINA procedure, but this is not as central a concern for PcGets, although Hendry often advocates reparameterizations of the GUM so as to strive for orthogonal regressors.

G. Subsamples

PcGets	RETINA
a. Two overlapping subsamples. b. Used only to check the significance of every variable in the final model to check the reliability of the selection.	a. Three disjoint homogeneous subsamples (essential feature of the procedure). If doubts arise as of possible clustering of the data in cross-section studies (say from preliminary data analysis), the observations may be scrambled. b. Used in order to cross validate out-of-sample using fresh data. c. The order of estimation, testing and cross validation is inverted and results are contrasted against one another.

In PcGets two overlapping subsamples are used for post-choice evaluation, but not for model selection. They are used to check for in-sample-goodness of fit. However, the model building and selection mechanism of RETINA is directed toward out-of-sample predictive ability. That is why it is necessary to use disjoint subsamples for performing out-of-sample forecasts and carrying out truly out-of-sample forecast evaluations.

H. Explanatory variables

PcGets	RETINA
Original variables and transformations	Original variables and nonlinear

specified in the GUM.	transformations allowed for by the procedure.

I. Linearity

PcGets	RETINA
a. Linear or nonlinear in the parameters, as specified by the GUM. b. Linear or nonlinear in the underlying variables, as specified by the GUM	a. Linear in the parameters b. Linear or nonlinear in the underlying variables.

The linearity or nonlinearity in the underlying variables afforded by PcGets is specified by the GUM. The nonlinearity in the underlying variables of the model selected by RETINA is chosen by the procedure. Linearity in the parameters ensures simple computation.

J. Functional form

PcGets	RETINA
Assumed that can be approximated by a model nested in the GUM	Assumed that it can be approximated by the allowed transforms

The functional form of the model suggested by PcGets is embedded in the GUM. The functional form of the model suggested by RETINA is given by the original inputs and the transforms allowed by the procedure.

K. Types of data applicable so far

PcGets	RETINA
Time series or cross section	Mainly cross section at present (no obstacles to its application in a time series context).

L. Algorithms behind PcGets and RETINA

PcGets	RETINA
a. Specify the GUM, based on theory, seeking a relatively orthogonal specification. b. Estimate the GUM on the whole sample. c. Select the set of mis-specification (residual and parameters) tests. Reduce the	a. Select a candidate set of potentially predictive variables (inputs) b. Generate the selected transformations of the inputs. Select three homogeneous disjoint subsamples. Order the transforms by a saliency

GUM by repeated in-sample testing. Obtain a baseline model. d. Multiple reduction path searches to obtain terminal selections. e. Repeat the above using as GUM the union of the terminal models. f. The significance of every variable in the final model is assessed in 2 over-lapping subsamples to check the reliability of the selection.	feature. c. Select and estimate on subsample 1 a base model adding the transformations one by one in the order of the saliency feature subject to a collinearity constraint. d. Cross-validate out-of-sample the previous model using subsample 2. e. Repeat the above, estimating the previous model in subsample 2 and cross-validating out-of-sample in subsample 3. Use out-of-sample AIC. f. Repeat steps a. through d. for all 6 combinations of the three subsamples. g. Select one of the 6 previous models by estimating each of them in 2 of the subsamples and cross validating in the third.

Undoubtedly, the two procedures have certain parallel features. Both apply repeated testing: PcGets reduces the GUM from the top down with in-sample tests while RETINA builds from the bottom up by adding regressors one by one, in the order suggested by a saliency feature, controlling for collinearity, and then repeatedly using out-of-sample tests and an information criterion, the out-of-sample AIC, for trimming the model down.

Despite these parallel features, especially in the mechanics, PcGets and RETINA are nevertheless distinct in terms of objectives, general strategy, selection criterion, the base model, flexibility of the model, treatment of collinearity, and the use of the subsamples.

The procedures can be seen as different but far from incompatible ways of providing insights into a set of data. Different emphases are at work and, in particular applications, one may be more useful than the other, depending on prior knowledge, the intended use of the model, and the ability of the researcher.

4. RETINA and PcGets: an application.

The aim of this section is to illustrate the performance of RETINA and PcGets using a real data set. The comparison is necessarily limited in scope, since the GUM is crucial for PcGets, and its specification is crucial for a proper comparison with other methods. We do not have clear indications as to what might be legitimate GUMs in our particular application, so we use simple models to start with and feed the same inputs to the linear regression model, RETINA and PcGets. By the same token, not having strong hypotheses about the function linking the available information is a fundamental basis for the use of RETINA.

For our application we use a cross section of US firms. The data were obtained from PNR, a subsidiary of Indetec International. The data set includes variables related to the demand for business toll telephone services in 1997. Our main interest is the duration of "intra-lata" calls. (LATA stands for "Local Access and Transport Area.") A toll call from one point within a LATA to another point within the same LATA is an intra-lata or short distance toll call. Possible explanatory variables of intra-lata duration are: business lines [Bus], the number of hunting lines [Hun], the sales [Sales] of the company expressed in dollars, the number of employees working locally [Emh], the total number of employees for the business [Emt], the physical extension of the business, proxied by the square footage of its premises [Sqft], and population [Pop], that is, the size of the business area location.

Our sample has 1217 observations. Prior to the analysis, the data were rescaled, to avoid the eventual negative impact on computations of large differences in orders of magnitudes of the variables. Given our purposes here, we focus on the performance of RETINA with respect to the more traditional use of a linear model. We do not consider prior transformations of the original variables, although a log transform of the dependent variable or its ratio by the number of employees (which is common in the Telecom demand literature) could also be considered (which also raises the issue of what transformation for the dependent variable should be considered).

We model intra-lata minutes as a function of the number of business lines, the number of hunting lines, employees here, employees total, sales, square footage and population habitat size.

To evaluate the candidate models, we use two criteria. The first is based on an Information Criterion; specifically, we choose the model with the lowest *AIC*. The second is the out-of-sample performance, measured by Cross-Validated Mean Square Prediction Error *(CMSPE)*, which we expect to be lower for models suggested by RETINA. To compute a consistent measure of *CMSPE*, we used the following strategy, motivated by the fact that *CMSPE* depends on the specific subsample split considered, in terms of its size and characteristics.

We randomly assigned each observation to three disjoint sub-samples, each including approximately one third of the observations in the sample. Then the proposed models were cross-validated using two of the three subsamples for estimation and the third for cross-validation. We considered all the three possible rotations and then summed the *CMSPE*'s obtained in each rotation.

Table 2. PcGets and RETINA: Selected Simple Models

	(1) Benchmark Linear Model	(2) PcGets (Liberal Strategy)	(3) PcGets (Conservative Strategy)	(4) RETINA Model 1 Original Inputs
CMSPE	**909.88**	**896.11**	**903.08**	**770.86**
AIC	**5.443**	**5.440**	**5.446**	**5.459**
Adj. R^2	**0.603**	**0.603**	**0.600**	**0.595**
Number of parameters	**8**	**6**	**5**	**5**
CONSTANT	-3.945 (.717)	-3.907 (.678)	-3.278 (.649)	-4.086 (.684)
Bus	2.508 (.207)	2.492 (.183)	2.579 (.182)	2.571 (.185)
Hun	.089 (.031)	.091 (.030)		.083 (.030)
Sales	-7.972 (41.852)			
Emt	.472 (.093)	.472 (.093)	.458 (.093)	
Emh	.930 (.132)	.933 (.130)	.921 (.131)	1.438 (.085)
Sqft	.450 (.061)	.450 (.061)	.462 (.061)	.481 (.061)
Pop	.033 (.184)			

Standard errors in parentheses.

In Table 2 above, we compare several model estimations. In column 1, we report the estimation of a usual linear model (by OLS) without using an automatic model selection strategy. The signs of the coefficients are as expected, except for Sales and Population, which are not statistically significant. The results suggest that the number of business lines [Bus], labour force [Emh], [Emt], and physical extension of the business [Sqft] should be important for explaining the duration of intra-lata calls.

In columns 2 and 3 we report the results of PcGets taking the standard linear model as the GUM. The liberal strategy drops the two insignificant variables of the basic linear model, and obtains a slightly lower *AIC* and a better *CMSPE*, providing an improvement over the basic model. The conservative strategy drops an additional variable, hunting lines [Hun], with a slight increase in AIC and worsening of the *CMSPE*. In terms of prediction, it is still better than the basic model of column 1.

In column 4, we report the model suggested by RETINA without allowing any transformation of the inputs. We do this to obtain a direct comparison with the basic linear model and with the models suggested by PcGets. We observe that the first model suggested by RETINA - model 1, is very similar to the conservative PcGets model. The difference is that it substitutes [Hun] for [Emt].

The comparison of columns 2 and 4 illustrates how RETINA has improved out-of-sample predictive ability (corresponding to an improvement of the CMSPE from 896.11 to 770.86) by dropping the variable Emt even if it is highly significant in sample, with a *t*-statistic of 5.07, a result which parallels the outcomes in the simpler models seen in Table 2. If one had a strong prior to the theoretical relevance of such a variable the indications provided by the automated procedure should definitely be taken into account.

In this case, RETINA chooses just one of the two variables related to Employees in the model, as the number of workers locally and in the whole business is highly correlated (Pearson’s correlation=0.86). On the other hand,

it excludes more descriptive and non-significant variables such as Sales and Population.

Table 3 about here

A distinguishing feature of RETINA is its use of nonlinear and interactive transforms of the underlying predictors. Accordingly, in Table 3 - columns (1) and (2) - we report the results obtained by RETINA, permitting it the use of nonlinear transforms and interactions created by taking all squares, cross-products, and cross ratios. For brevity, we do not explicitly reference all such ratios as most of these are never selected. As a result the out-of-sample (as well as the in-sample) performance improves significantly relative to the results of Table 2, while at the same time being only slightly less parsimonious than the simple linear model of Table 2. RETINA Model 2 (column 1) was chosen using a minimum *AIC* criterion and RETINA Model 3 (column 2) considering the minimum *CMSPE*.

The total variation explained by these models is noticeably higher than models with the original inputs, whereas the *CMPSE*s for models 2 and 3 are respectively 37% and 43% lower than that of the first basic model (column 1).

The transformations suggested by RETINA for both models 2 and 3, in particular the interaction terms, are interesting. The negative constant terms in columns 1-4 become positive in columns 5 and 6, in which RETINA allows for transformed inputs. These models are not necessarily correctly specified, and we emphasize that this implies that the coefficients thus do not necessarily have the standard *ceteris paribus* interpretation.

Although PcGets is designed to identify a theory-congruent parsimonious model corresponding to the true DGP, it may nevertheless have value in obtaining forecasts. To investigate the properties of PcGets in this regard in a manner directly comparable to our RETINA results, we apply PcGets by specifying a GUM that contains all the transforms accessible to RETINA. Strictly speaking, this is a non standard approach to PcGets, as we have no theory justifying this specification of the GUM, nor a guarantee that the result may be theoretically plausible. We may be placing theory congruence at risk,

and we make no attempt to specify relatively orthogonal regressors. Nevertheless, we see from columns (3) and (4) of Table 3 that both conservative and liberal versions of PcGets not only provide noticeable improvements in CMSPE and AIC relative to all the models of Table 2, but also improvements relative to the RETINA results of column (1) and (2) of Table 3 (CMSPE of column (3) of PcGets is 3.7% lower than column (2) of RETINA).

That PcGets is useful in this context should not, however, be surprising, considering that (as is plausibly true here) even when the GUM does not include the DGP (the ideal forecasting tool), PcGets should identify an approximation to the DGP that has certain optimality properties and that may thus be useful in forecasting. Interestingly, although PcGets is designed to deliver parsimonious models, it achieves its modest gains relative to RETINA with models that are twice as complex.

We emphasize the strictly limited scope of the above comparison. Nevertheless, our example illustrates that RETINA may help the researcher not only to arrive at useful forecasting models, but also to consider relevant transformations of the inputs, that, when used as regressors, yield models with improved forecasting ability relative to the basic regression model. Further, we see that PcGets has value in obtaining useful forecasts. It is an interesting question to investigate whether the use of PcGets for this purpose generically results in less parsimonious models than RETINA or whether this is only a feature of our present example

5. Conclusions.

Our discussion has revealed both differences and similarities between the two automated modelling procedures, PcGets and RETINA. Although some polarization of reality should be discounted in what follows, as matters are not always so clear cut, the main differences between the two approaches concern:

a. **Objectives**: In PcGets, obtain an appropriate representation of the data within sample (with all caveats about implicit assumptions often

encountered in the literature that the true DGP is among the models being considered) or, in RETINA obtain a predictive approximation.

b. **Base model:** GUM specified by the researcher prior to the use of PcGets or constructed within the procedure from the relevant transformations of the inputs in RETINA.
c. **General strategy:** general-to-specific in PcGets or specific-to-general with later reduction as in RETINA.
d. **Selection Criterion**: in-sample specification tests (residuals and parameters) in PcGets or out-of-sample predictive performance, in RETINA (cross-validation, no specification tests).
e. **Flexibility:** embedded in the GUM and later selected by the reduction process in PcGets or chosen by the procedure in RETINA from the allowed transformations.
f. **Collinearity**: checked for in PcGets when constructing the GUM or explicitly embedded checks in the procedure as in RETINA (with some caveats in order since RETINA does not recognize the meaning of the variables involved).
g. **Subsamples:** used as a final post-selection check in PcGets or an essential part of the procedure in RETINA.

The complementarities of both approaches are also important:

1. PcGets may be more appropriate when there is a strong desire to conform to a theory, or reasonable confidence that the *GUM can be seen as a good representation of the DGP* (although there is no formal assumption that the DGP is nested within the GUM in PcGets, practitioners often do operate as if the DGP is among the models considered in the selection); RETINA may be more appropriate when the researcher *lacks precise knowledge* of the relationship between the dependent variable and the inputs. The role of the GUM is crucial in PcGets; however, it is generated outside PcGets.
2. PcGets may be more useful when an *in-sample* fit is desired, whereas RETINA may be more appropriate when the objective is an *out-of-sample* predictive performance. Nevertheless, Pc GETS has the potential to identify useful forecasting models when provided with a sufficiently flexible GUM, as our example shows.

3. In case of *high collinearity* between the original inputs or the regressors of the GUM, RETINA may be the procedure of choice, since it explicitly embeds the control of collinearity, although in a time series context RETINA as currently implemented would not be able to guide between the addition of, say, a lagged variable or a variable in first differences. Moreover, the degree of collinearity among regressors depend on the way variables are originally included in the information set: it is certainly true that models containing, say X1 and X2 on the one hand and X1 and Z=X2-X1 are equivalent, but the degree of collinearity between X1 and X2 is different than that of X1 and Z (or X2 and Z for that matter).

By and large, while it seems that a reasonable approach would be the creation of a "RetiGets" hybrid, one should bear in mind some facts:

a. PcGets is mostly designed for macroeconomic aggregate time series data, whereas RETINA has been developed for cross section individual data of economic or non-economic nature.
b. PcGets is more inclined towards modest size samples and number of regressors, whereas RETINA has been developed for large samples and possibly large numbers of inputs.
c. PcGets may be more useful when an in-sample fit is desired whereas RETINA may be more appropriate when the objective is an out-of-sample predictive performance.

It is apparent that both procedures are still at an early stage of development, especially RETINA. In particular, these methods can be extended to handle other types of data and estimation techniques: e.g. models with stationary and nonstationary variables among the candidate regressors, panel data, limited dependent variable models, systems of equations; and their performance in these contexts needs to be assessed. Nevertheless, the usefulness of these automatic modelling methodologies so far certainly warrants these further developments.

References.

Campos J., Ericsson, N.R., and D.F. Hendry (2004). *Readings in General-to-Specific Modeling*, Edward Elgar, Cheltenham, forthcoming.

Hendry, D.F. and Krolzig, H.-M. (2001) *Automatic Econometric Model Selection with PcGets.* London. Timberlake Consultants Press.

Hendry, D.F. and Krolzig, H.-M. (2003). 'New Developments in Automatic General-to-Specific Modeling', in B.P. Stigum (ed.), *Econometrics and the Philosophy of Economics*, Princeton University Press, Princeton.

Hendry, D.F. and Krolzig, H.-M. (2004). 'Sub-sample Model Selection Procedures in Gets Modelling', forthcoming in R. Becker and S. Hurn (eds.), *Advances in Economics and Econometrics: Theory and Applications*, Edward Elgar, Cheltenham.

Hoover, K. and Perez (1999) Data mining reconsidered: Encompassing and the general-to-specific approach to specification search, *Econometrics Journal*, 2, 167-191.

Krolzig, H.-M. and D.F. Hendry (2001). 'Computer Automation of General-to-Specific Model Selection Procedures', *Journal of Economic Dynamics and Control*, 25, 831-66.

Pérez-Amaral, T., Gallo, G. M. and H. White (2003) 'A Flexible Tool for Model Building: the Relevant Transformation of the Inputs Network Approach (RETINA)', *Oxford Bulletin of Economics and Statistics*, 65 (s1), 821-838.

Ploberger, W. and Phillips, P.C.B. (2003) Empirical Limits for Time Series Econometric Models, *Econometrica* 71 (2), 627-673.

White, H., (1989). 'Learning in Artificial Neural Networks: A Statistical Perspective', *Neural Computation*, Vol. 1, 425-64, (reprinted in White, H. (1992). *Artificial Neural Networks: Approximation and Learning Theory*. Oxford, Blackwell).

White, H. (1998). 'Artificial Neural Network and Alternative Methods for Assessing Naval Readiness'. Technical Report, NRDA, San Diego.

Table 3. PcGets and RETINA: Selected Flexible Models. Transformed Inputs

	(1) RETINA Model 2	(2) RETINA Model 3	(3) PcGets (Conserv.)	(4) PcGets (Liberal)
CMSPE	**572.01**	**518.00**	**498.63**	**507.42**
AIC	**4.932**	**4.947**	**4.839**	**4.839**
Adj. R^2	**0.757**	**0.756**	**0.784**	**0.785**
# of parameters	**9**	**9**	**18**	**19**
CONSTANT	3.240 (.633)	3.792 (.425)	3.760 (.532)	3.196 (.644)
Bus	1.116 (.182)			
Hun			-.263 (.056)	-.269 (.058)
Emt	-.446 (.097)			
Sqft	-.461 (.111)			
Pop			2.949 (.664)	3.204 (.693)
Bus^2		.842 (.117)	1.125 (.153)	1.151 (.154)
Hun^2			.397 (.130)	.326 (.129)
$Sqft^2$	.080 (.033)			
Pop^2			-.318 (.087)	-.333 (.089)
$1/Pop^2$				.116 (.036)
Hun * Emh	1.193 (.060)	1.328 (.076)	1.176 (.095)	1.155 (.096)
Bus * Emh			2.706 (.442)	2.783 (.441)
Hun * Sqft			-.098 (.025)	
Bus * Sqft	1.042 (.116)	1.102 (.091)	1.210 (.099)	1.212 (.099)
Emt * Sqft	.485 (.047)	.415 (.040)	1.233 (.148)	1.262 (.148)
Emh * Sqft	-.449 (.039)	-.650 (.049)	-1.181 (.182)	-1.213 (.181)
Emt * Pop		-.139 (.043)		
Sqft * Pop		-.047 (.018)	-.092 (.020)	-.108 (.020)
Bus * Emt			-2.854 (.411)	-2.931 (.410)
Bus * Pop			-.287 (.057)	-.275 (.057)
Hun * Pop			.524 (.104)	.457 (.106)
Emh / Pop			-.424 (.086)	-.445 (.095)
Hun / Pop			.364 (.095)	.261 (.105)
Sqft / Pop				-.598 (.169)
Hun * Sales		-.689 (.264)		

Standard errors in parentheses.