HYBLOCK: A ROUTINE FOR EXPLORATORY FACTORING OF BLOCK-STRUCTURED DATA

William W. Rozeboom

University of Alberta

ABSTRACT

Suppose that you have empirical data on variables that include multiple indicators for one or more blocks of hypothesized source factors on which your model imposes a causal-path structure without specifying the number of factors in each block. Here is an EFA (Exploratory Factor Analysis) way to solve the data covariances for simple-structured oblique factors conforming to your model's block paths without need to impose the additional constraints required for a SEM (Structural-Equations Modeling) solution.

Alternatively, if you don't care much for causal-path modeling, Hyblock results also have a generalized marker-variable interpretation that you may find more congenial.

Key words: Factor analysis, causal-path modeling, subspace-constrained rotation

Last revised: November 1998

HYBLOCK: A ROUTINE FOR EXPLORATORY FACTORING OF BLOCK-STRUCTURED DATA

Wm. W. Rozeboom

University of Alberta

Overview.

Envision any more or less orthodox acyclic causal-path model, with a circle for each latent variable and a square for each manifest one. Connect these nodes by causal-path arrows → under the constraints (a) that no forward-continued path makes a closed loop so that → is a partial-order relation, (b) that each latent variable (common source factor) is directly antecedent to at least one manifest variable (data item), and (c) that no manifest variable is an intermediate node on any path. Also take any mediated path from one node to another to imply a possible direct (unmediated) path from the first to the second as well. Solving the manifest-variables' covariances for the path coefficients in such a model is routine for any modern structural-equations modeling ("SEM") program such as Lisrel. But now, replace each circle with a block of source factors having unknown dimensionality, replace each square with a specified block of data variables, allow all dependent factor blocks to contain unexplained covariance, and stipulate that coefficient indeterminacies arising from permissible factor rotations are to be resolved by optimizing the prevalence of near-zero path weights. The Hyblock procedure described here determines (with some assistance from user discretion) the appropriate dimensionality of each factor block and from there solves for the model's path parameters and factor correlations without need for additional model constraints.

The result of Hyblock analysis is an oblique factor pattern wherein (a) nearly all of each data variable's common part lies in the space spanned just by factors in the blocks declared path-antecedent to it; and (b) the number of appreciably nonzero pathweights through which each data variable is determined by the assorted factors path-antecedent to it are minimized. In particular, when factor f_j is a direct source of item y_k and factor f_i path-precedes f_j, Hyblock diagnoses the extent to which the effect of f_i on y_k is mediated by f_j. (Of course, this diagnosis may be be imperfect; but that uncertainty afflicts all latent-variable models.)

Note: I write names for the programs in my DOS-operated Hyball factor-analysis package in capital letters, e.g. PROG (not an actual Hyball program), while a procedure centered on PROG but involving other programs as well is called "Prog" with only the first letter capitalized. What makes the Hyblock procedure described here most distinct from other varieties of exploratory factoring is fixation of nested factor-block subspaces by Hyball program HYBLOCK. This has been designed to intervene between initial factor extraction by Hyball program MODA (short for "Multiple Output Dependency Analysis") and rotation to simple structure under those rotation constraints by HYBALL (short for "HYperplane eyeBALLing"). But HYBLOCK can also be entered with extraction patterns imported from data-analysis packages other than Hyball.

Rationale.

The block-structured path model.

Factoring by Hyblock is appropriate for covariance structures whose empirical variables have been sorted into blocks corresponding to hypothesized blocks of source factors on which a causal-path structure is supposed. (An alternative interpretation that avoids construing Hyblock's path structure as causal dependency will be described later.) Specifically, the data covariances are to be modeled as follows:

1. Partition the total array $Y = \langle y_1,...,y_m \rangle$ of data variables into $r \geq 2$ disjoint blocks $Y_0, Y_1,...,Y_r$, where Y_0 is a possibly-empty set of manifest inputs (observed independent variables viewed as exogenous sources) such as experimental-treatment contrasts. The number N_{Y_k} of variables in block Y_k ($k \geq 1$) can be as low as 1, though more is clearly preferable.

2. View each data block Y_k ($k = 1,...,r$) as comprising more-or-less noisy manifestations of a block F_k of common factors whose to-be-determined dimensionality N_{F_k} has bounds $0 \leq N_{F_k} \leq N_{Y_k}$. (Variables Y_k are understood to be primarily indicators of the factors in block F_k, but perhaps not exclusively of those.) If Y_0 is not empty, $F_0 = Y_0$ is a default stipulation that can later be overridden by setting $N_{F_0} = 0$ and relocating the manifest inputs elsewhere in the path structure.

3. Posit a structure of causal-path dependencies on factor blocks $F_1,...,F_r$ extended to include each indicator block Y_k on an output path from F_k. (Each Y_k and dependent F_k is of course allowed to include residual variance unaccounted for by its modeled path antecedents.) This path structure must be a strict partial order (transitive and anti-reflexive) that Hyblock requires without loss of generality to be embedded in the linear order of block indices $1,...,r$. Specifically, for each pair $\langle F_j, F_k \rangle$ of factor blocks with $j < k$ (the program will not accept $j \geq k$), HYBLOCK's user stipulates whether the model includes a direct causal path $\rightarrow$ from F_j to F_k; while if both $F_i \rightarrow F_j$ and $F_j \rightarrow F_k$ then also $F_i \rightarrow F_k$. (HYBLOCK automatically expands user-entered block dependencies to include all their transitivity entailments.) If $F_i \rightarrow F_k$ in this imposed path structure, that is, if F_i is path-antecedent to F_k, we shall say that block F_i is a *mediated* source of block F_k if also $F_i \rightarrow F_j \rightarrow F_k$ for some third nonempty factor-block F_j, or is an *immediate* source of F_k otherwise.[1] Finally, the model also presumes $F_0 \rightarrow F_k$ for all $k = 1,...,r$ unless F_0 is empty,[2] and stipulates that each block Y_k of indicator variables is causally dependent on factor block F_k. (Conceptually, F_k comprises whatever recoverable factors most immediately underlie the indicators in Y_k; so $F_k \rightarrow Y_k$ is mainly true by definition.) When F_k is empty, all factor blocks declared to be immediate sources of F_k are immediate sources of Y_k. By default, though not obligatorily, all factor blocks are on paths to the last (most dependent) data-block Y_r.

   Stipulation $F_j \rightarrow F_k$ in a Hyblock model is molar notation for positing a direct molecular path from each factor in F_j to each factor in F_k. But Hyblock also recognizes that many of these molecular paths from one factor block to another may well have negligible weight, and hopes to reveal the ones of which this is true. Note that when f_i is in a factor block taken by the molar model to be an F_j-mediated source of a factor block containing f_k, the model implies not merely a direct molecular path from f_i to f_k but also (generally) a spindle of indirect f_i-to-f_k paths severally mediated by the various factors in F_j. (We mention this webwork of mediated molecular paths only for background clarification; it plays no role in Hyblock's explicit concerns.) Happily, this complexity is nicely domesticated by our classical presumption of causal linearity: The strengths of molecular path immediacies $\{f_j \rightarrow f_k\}$ are measured by scalar coefficients in linear equations, from which the force of a direct molar path $F_i \rightarrow F_k$ between factor blocks is described by a matrix of the direct molecular pathweights from factors in F_i to factors in F_k. And you are already familiar with the power and beauty that matrix algebra brings to models such as this.

4. Presume for all $k = 1,...,r$ that the data covariances in indicator block Y_k are essentially due just to factors in F_k as well as, perhaps, in other blocks on which F_k is declared dependent. (The qualifier "essentially" here acknowledges that although the model solution attempts to fit this premise, its success is unlikely to be perfect.) And presume also that causal linkage in this system of variables is *frugal*, meaning that a good proportion of coefficients in the model's path-weight matrices should be negligible. In particular, when F_j is a mediated source of Y_k, most if not all the direct path weights from F_j to Y_k are expected to vanish when the factor axes are properly positioned. By attempting to optimize fit to these desiderata through admissible rotation of axes after the subspaces spanned by the block factors have been identified, Hyblock endeavors to *discover* the detailed path structure of these data within the posited block ordering.

Computational Theory.

The structural equations that HYBLOCK solves are actually quite simple, rather more so than the foregoing inventory of premises may lead you to expect. For each block Y_k of dependent data variables, we seek to find coefficient matrices $\mathbf{A}_k$, $\overset{*}{\mathbf{A}}_k$, and diagonal $\mathbf{D}_k$ such that Y_k's covariances within and between all data blocks can be explained to a high degree of approximation by presuming Y_k to have structural composition

[1][3] $$Y_k = \mathbf{A}_k F_k + \overset{*}{\mathbf{A}}_k \overset{*}{F}_k + \mathbf{D}_k U_k + noise \qquad (k = 1,...,r)$$

where F_k is the block of Y_k's direct sources in the path model, $\overset{*}{F}_k$ is the union of all factors in the model's mediated Y_k-sources, i.e. of all blocks path-antecedent to F_k, U_k comprises normalized unique factors orthogonal to all other common and unique factors throughout the model, and *noise* is approximation error. (When Y_0 is not empty, equations [1] are provisionally expanded to include case $k = 0$: $Y_0 = F_0$). The path model also implies that F_k too has a structural determination

$$[2] \qquad F_k = \overset{*}{\mathbf{A}}_k \overset{*}{F}_k + G_k \qquad (k = 1,\ldots,r)$$

in which G_k is some composite of exogenous F_k-sources. (In fact, it will become plain that only factors with appreciable residual variance in G_k can be recovered in block F_k.) However, [2] needn't be considered when solving for the weights in [1]; and although Hyblock also produces a solution of [2], the auxillary assumption behind should not be presumed robust.

Operationally, Hyblock passes through three stages of computation, or four if you count initial computation of data covariances from raw scores. Each stage terminates with an archivable output file which delivers input to the next stage when you are ready to proceed.

Stage 1 (executed by Hyball program MODA): First, given the covariances $\mathbf{C}_{YY}$ (presumably standardized as correlations) among data variables $Y = \langle y_1,\ldots,y_m\rangle$, solve for a traditional reduced-rank estimate $\tilde{\mathbf{C}} \approx \mathbf{C}_{YY} - \mathbf{D}_u^2$ of the covariances among the data variables' common parts $\tilde{Y} = \langle \tilde{y}_1,\ldots,\tilde{y}_m\rangle$, where $\mathbf{D}_u^2$ is a diagonal matrix of uniquenesses (constrained to zero for manifest inputs) whose diagonal blocks comprise your solution for $\{\mathbf{D}_k^2\}$. It does no harm and indeed is generally beneficial to extract appreciably more initial factors than conventional exploratory factoring would approve--excess that proves unwanted can be shed later. (That is, Hyblock is indulgent of overfactoring.) Whatever rank N_R you accept for $\tilde{\mathbf{C}}$ provisionally becomes the total number $N_F = \sum_{k=0}^{r} N_{F_k}$ of common factors $F = \langle F_0, F_1,\ldots, F_r\rangle$ (N_{F_k} the dimensionality of F_k) you aim to recover. And replacing the data variables in model equations [1] by their common parts relative to these extraction factors simplifies [1] to

$$[3] \qquad \tilde{Y}_k = \mathbf{A}_k F_k + \overset{*}{\mathbf{A}}_k \overset{*}{F}_k \qquad (\tilde{Y}_k =_{\text{def}} Y_k - \mathbf{D}_k U_k - \textit{noise})$$

while enabling reproduction of $\tilde{\mathbf{C}}$ by common-parts model [3] to be exact. More importantly, since the rank N_R of $\tilde{\mathbf{C}}$ is less than the dimensionality of $\tilde{Y}$, *each factor block is some linear combination* $F_k = \mathbf{W}_k\tilde{Y}$ *of the common-part variables whose covariance estimate* $\tilde{\mathbf{C}}$ *is numerically in hand.*

Stage 2 (executed by program HYBLOCK after loading the extraction pattern): Taking advantage of the rotational indeterminacies in [3], HYBLOCK next chooses an initial axis placement satisfying the path model under which solution for the $\mathbf{W}_k$ in

$$F_k = \mathbf{W}_k\tilde{Y} \qquad (k = 0,1,\ldots,r)$$

is transparent. You fix these initial factors (i.e. identify $\mathbf{W}_k$) recursively, in order of the block indices. And because the block indexing has insured $j < k$ whenever $F_j \rightarrow F_k$, all factors in blocks path-antecedent to F_k are fixed before you undertake fixation of F_k.

Suppose that you have identified the coefficients in $F_j = \mathbf{W}_j\tilde{Y}$ for all $j = 1,\ldots,k$-1, and now prepare to fix F_k. For any choice of F_k that satisfies [3], partialling $\overset{*}{F}_k$ out of F_k yields other choices of F_k that also satisfy the model and from which, together with $\overset{*}{F}_k$, any other admissible F_k can be reclaimed later, namely, any rotation and rescaling of the F_k-residual. So you are free to stipulate that in your initial axis placement, F_k is orthogonal to $\overset{*}{F}_k$. Then $\overset{*}{\mathbf{A}}_k$ comprises the coefficients of

$\tilde{Y}_k$'s regression upon $\overset{*}{F}_k$, and is directly computable because $\overset{*}{F}_k$ contains only factors already fixed in $\tilde{Y}$-space. So the matrix $\mathbf{C}_{E_k E_k}$ of covariances among common-part residuals

$$E_k =_{def} \tilde{Y}_k - \overset{*}{\mathbf{A}}_k \overset{*}{F}_k = \mathbf{A}_k F_k$$

is also computable. Finally, the indeterminacy still remaining in your initial specification of F_k can be resolved by declaring F_k to comprise the principal axes of E_k corresponding to the N_{F_k} eigenvalues of $\mathbf{C}_{E_k E_k}$ that you consider appreciable. This identifies $\mathbf{A}_k$ from the eigenstructure of $\mathbf{C}_{E_k E_k}$, while $\mathbf{W}_k$ is left-inverse $(\mathbf{A}_k' \mathbf{A}_k)^{-1} \mathbf{A}_k'$ of $\mathbf{A}_k$--thus completing step k of your initial model solution and providing the information about $\overset{*}{F}_{k+1}$ needed for solution-step $k+1$.[4]

After final Stage-2 solution step $k = r$ has been completed, the total number N_F of common factors you have partitioned among factor blocks $F_1, \ldots, F_r$ may well be less than the number N_R of factors in the extraction pattern, especially if you overfactored generously. When this occurs, HYBLOCK creates an additional array of *Waif* factors comprising the normalized principal components of the data parts in initial extraction space orthogonal to the factors you have judged strong enough to retain in one or another of the F_k. These Waifs are included in HYBLOCK's output to Stage-3 rotation, and can either be dumped or processed further when that commences. But before HYBLOCK exits, information it displays on the distribution of residual Waif variance may suggest some revision of the dimensionalities you have allocated to your factor blocks. (More on this later.)

Stage 3 (executed by program HYBALL on the block-structured pattern received from HYBLOCK): Finally, when factor blocking is complete, you want to relax your temporary stipulation that each F_k in [3] is orthogonal to $\overset{*}{F}_k$. The permissible alternatives are surveyed by oblique rotations of factor totality $F = \langle F_0, F_1, \ldots, F_r \rangle$ under the constraint that each rotated F_k must remain in the subspace spanned jointly by the original F_k and $\overset{*}{F}_k$. HYBLOCK delegates this search by passing the pattern on the initial F, together with the block structure and the F-covariances (which are orthogonal only where path-connected[5]), to routine HYBALL. This translates the path links and factor-block memberships (that is, which factors belong to what blocks) into control parameters that keep each rotated factor within its assigned subspace, and rotates the full pattern to oblique simple structure under these subspace constraints. The result is a solution wherein overall molecular-path connectivity is minimized. More precisely, it is only the direct molecular paths from factors to data variables (in SEM jargon, the "measurement model") that HYBALL makes sparce. Although molecular paths between connected blocks could with some programming effort also be included in this overall minimization, the problematic accuracy of their estimation (see immediately below) makes that imprudent.

In addition to reporting the Y-variables' loadings on the rotated factors, HYBALL also returns the regression of each rotated factor block F_k upon its path antecedents $\overset{*}{F}_k$ together with the residual F_k-covariances unaccounted for by $\overset{*}{F}_k$. If the exogenous F_k-sources composing or producing G_k in [2] are orthogonal to $\overset{*}{F}_k$, the regression weights so computed are the molecular path coefficients of $\overset{*}{F}_k$ for F_k in [2]. But insofar as part of $\overset{*}{F}_k$'s correlation with F_k derives from their mutual correlation with exogenous sources of F_k--and you must presume that likely to at least some modest extent--this regression estimate of $\mathbf{B}^*_k$ may well be seriously distorted. In principle the covariances you obtain among the rotated factor blocks can be analyzed in depth considerable greater than this simple regression estimate of $\mathbf{B}^*_k$, but only for factor blocks with N_{F_k} large enough for meaningfully factoring.[6]

Subjectivities.

Although Hyblock has been described algorithmically above, the procedure includes considerable user involvement. For openers, the data-block groupings and their path structure are strictly user decisions: If your theory of the data doesn't tell you precisely what these should be, the program will not decide this for you albeit its results for a particular choice may urge you to re-consider that. Further, each stage of the analysis is interactive in some major respect beyond method minutia such as convergence criteria and iteration limits that seldom matter much. The subjectivities in Stages 1 and 3 are old acquaintences in exploratory factoring: Stage 1 requires you to decide what total common-factor dimensionality to accept, although as already noted Hyblock tolerates considerable initial overfactoring insomuch as Waifs can later bleed off excess. And Stage 3 grapples with the classic problem of rotation indeterminacy. HYBALL has a keen eye for detecting hyperplanes; but these may well be ambiguous in your data, and in that case you may need to run many variations on HYBALL's rotation options before you are satisfied that you have found the best permissible axis placement.

The most troublesome Hyblock subjectivities, however, are likely to be your Phase-2 decisions about block dimensionalities $\{N_{F_k}\}$. Ideally, the eigenvalue curve for each $\mathbf{C}_{E_kE_k}$ will either remain comfortably large throughout, in which case $N_{F_k} = N_{Y_k}$ is appropriate and creates no problem if N_{Y_k} is quite small, or drops abruptly to vanishing from a level clearly too large to ignore. But if the eigenvalues for this block subside gradually to near-zero while N_{Y_k} exceeds what can be afforded for N_{F_k}, the common variance in data block Y_k cannot be entirely explained just by factors in the path stipulated for this, and you must decide somewhat arbitrarily by your selection of N_{F_k} less than N_{Y_k} how much of $\mathbf{C}_{E_kE_k}$ to abandon to off-path factors.[7] You can and indeed should experiment with these choices; but to the extent they seem arbitrary with an appreciable amount of E_k-variance in the range of uncertainty, rather than demanded by a sharp eigenvalue drop to negligible residuals, the initial solution for $\tilde{\mathbf{C}}$ cannot be viewed as fitted cleanly by the path structure you have imposed on [3].

To be sure, having chosen $\{N_{F_k}\}$ with arbitrary cuts on the eigenvalue curves for one or more factor blocks, you can zero out all the off-path pattern coefficients in the completed solution for initial F, use the purified pattern together with the computed F-covariances to reconstruct the common-parts covariance matrix, and judge how inferior the latter is to $\tilde{\mathbf{C}}$ for approximating $\mathbf{C}_{YY}-\mathbf{D}_u$. (Hyball does not now implement such pattern purefication, but may do so in later releases.) You may--or may not--decide that the path-purified reconstruction is virtually as good as the solution for this same global N_F prior to imposing the path structure. Either way, the salient point is that here is where your path assumptions receive empirical testing. (Of course, even if the path model fits perfectly you have no assurance that its coefficients are truly directed causal weights. But that is another issue altogether.)

Exploiting the Waifs

As described above, Waif factors arise when HYBLOCK's provisional solution for block factors, that is, ones considered the most immediate sources of restricted item blocks, leave some dimensions of initial extraction space unassigned to any factor block in your posited path structure. Ideally, these are uninterpretable residues brought about by initial overfactoring. But in practice your Waif variance will often be larger than you can comfortably ignore. If so, there are two ways in which you can make use of what is salient in this, one when finishing HYBLOCK and the other at start of HYBALL.

First, before finalizing your choice of dimensionalities for factor blocks, you are invited to revise this in light of how Waif variance is distributed across item blocks. If you accept this option, HYBLOCK first rotates the Waif's principal axes to Varimax simple structure and displays for each rotated Waif its mean and number of loadings, separately in each item block, that are larger than a repetitively adjustable value *Cut*. By setting *Cut* to a level excluding all but the largest loadings on the most prominant Waifs, you can judge whether to try redirecting some of this Waif variance into block factors. In contrast, a pronounced lack of Waif variance in some block urges you check whether more factor dimensionality has been assigned to this than it really needs.

However, rotated Waifs with appreciable loadings in more than one item block often elude capture by accepting more of those blocks' principal factors. The place to salvage interesting Waifs is in HYBALL. No dimensions of the common-factor space extracted in Stage 1 are discarded in Stage 2; rather, all principal axes of the Waif residuals are appended to the block factors in HYBLOCK's Stage 2 output file. When HYBALL loads this to commence Stage 3 rotation, it first rotates the Waif axes to Varimax simple structure and displays information on their salient loadings sufficient for you to pick out any rotated Waifs you would like to preserve for further study. (The remainder are deleted from the input pattern, though you can always reload the HYBLOCK-output file to start again.) Any Waifs you retain are by default treated as "isolates" whose loadings are included in pattern displays but neither rotate further nor contribute to rotation of the other factors. Once you become familiar with Waif management, however, you can also insert these selectively into the HYBLOCK-defined factor blocks or otherwise allow them to participate in the ensuing oblique rotation to simple structure.

Operating details.

Hyblock is more difficult to use than an orthodox unstructured factoring procedure, but only because it is easy to become confused over what is where in the block structure. If this is at all complex, it is important for you to prepare a carefully indexed chart of relevant information to consult when need arises. Several sets of integers require coordination in this chart, starting with a list of indices from 1 to r for the factor blocks (disregard F_0 at this point even if $N_{Y_0} > 0$). Paired with each block index k, you want: (a) an index listing of the N_{Y_k} data variables in block Y_k; (b) an index listing of the factor blocks, if any, path-antecedent to F_k; and (c) optionally, an index listing of the N_{F_k} individual factors that are in this block. Listings (a) and (b) are needed as input to Stage 2, but can just as well be prepared before Stage 1. (For maximal convenience, your data variables should be so ordered that the items in each Y_k are indexed consecutively.) In contrast, listing (c) cannot be completed until the end of Stage 2, serves only to facilitate study of output from Stage 3, and can be copied from the block-structure table therein.)

Your preparation of this structure chart should commence with a graph containing names (brief verbal labels) for your blocks (nevermind at outset precisely what variables they contain) connected by arrows showing the direct dependencies you intend to impose on them. If your data include manifest inputs (nonempty Y_0), omit these from your planning graph if they are to be treated as sources of all factor blocks, but include a block for each if you intend to override this default path assignment for them. Next, index your block names consecutively from 1 to r in such fashion that each block's index is smaller than the indices of all blocks path-dependent on it. Once this ordering of blocks has been successfully completed, each block's index should be changed from numeral k to the kth letter α_k in alphabetic sequence A,B,...,α_r. That is, Block 1 becomes Block A, Block 2 becomes Block B, and so on. (This avoids confusion when entering block and item indices at keyboard.)

> To verify that your blocks are ordered correctly, or to help work this out if the complexity of your structure makes that difficult, call utility program ORDER and, when prompted, enter the index pairs $\{i,j\}$ for which you declare block i to be path-antecedent to block j. (For example, ORDER interprets entry "2 4, 3 2, 4 6" to signify path connections $2\rightarrow4$, $3\rightarrow2$, and $4\rightarrow6$. These pairs can be entered on one line, or more if you need that, with or without punctuation. Entailed path linkages needn't be entered.) ORDER expands your entries to make explicit all transitivities, and returns a detailed report on their structure, most importantly identification of any closed loops, and a left-to-right listing from least dependent to most dependent with brackets indicating most permissible permutations thereof. Your provisional block-index assignment should emerge from this ORDER listing in ascending sequence; if not, unless you have a closed loop the listing will show permissible index assignments from which you can take your pick. Loops you must break on your own.

Once your block names are suitably indexed, make a table with r rows consecutively labeled $\alpha_1,...,\alpha_r$ and containing three columns indexed by these block letters. Its first column receives in each row α_k the indices of the data variables in block Y_k(first and last suffice if these are indexed consecutively); the second lists the index letters of all factor blocks on which block F_k is directly dependent (additionally including indices of F_k's mediated source blocks is optional); and the third (optional) column awaits later insertion of the individual factor indices in F_k. (Factors 1 to N_{F_1} will be in block F_1, factors N_{F_1} to $N_{F_1}+N_{F_2}$ in block F_2, and so on in consecutive order.) Detailed organization of this table can be at your preference, and you may also wish to include the block name in each row.

Other than this preparation of a block-structure chart, fitting the Hyblock model to your Y-covariances procedes exactly like ordinary factor-extraction by MODA followed by HYBALL rotation (see Operating Instructions for the Hyball package) except for running HYBLOCK between exiting MODA and entering HYBALL. When called, HYBLOCK first invites you to select an initial factor pattern for variables Y from a displayed list of MODA's pattern solutions saved in your active directory and, if the selected pattern is found to contain manifest inputs (nonempty Y_0), allows you to make these explicit in the block structure rather than defaulting to block F_0 implicitly path-antecedent to all extracted factors. (How you respond should, of course, agree with the plan of your structure chart.) You next enter, for each factor block F_k in order $k = 1,\ldots,r$, first the indices of data variables Y_k diagnosing this block and then letter indices of whatever factor blocks are directly path-antecedent to block F_k. (These listings should be on display in your structure chart.) After the block structure is fully entered and approved, you again sequence through block indices $k = 1,\ldots,r$, this time to examine the eigenvalues of each residual covariance matrix $\mathbf{C}_{E_k E_k}$ and choose the number N_{F_k} of factors to put in block F_k. By the time you finish these $\{N_{F_k}\}$ assignments you may find yourself wishing you had chosen somewhat differently, especially if your generosity to the earlier blocks has left insufficiently many factors for justice to the later ones; so before exiting you are given opportunity to redo those choices after examining a simultaneous display of each block's residual eigenvalues and number of factors selected on this pass followed by Waif details. Once you accept your blocks' factor assignments, the program stores the pattern on these repositioned factor axes along with the block structure and factor covariances (which are orthonormal except between blocks not path-connected) in an unformatted HYBALL-input file, writes this same information to an ASCII see-file that you can view on screen or print if wanted, and stops. (Before these Stage-2 outputs are filed, you are also invited to accept permututation of your data variables into ascending block order corresponding to your factor indexing. This facilitates later interpretation of results, since your full pattern matrix then tidily partitions into submatrices pairing each indicator block with each factor block.) It then only remains to call HYBALL and rotate this initial HYBLOCK pattern just as you would any pattern received directly from MODA. HYBALL's option of revising received rotation constraints remains open; but since input from HYBLOCK defaults to rotation constraints that preserve the HYBLOCK-imposed block structure, overriding these will normally be unmotivated except perhaps for shifting promoting Waifs from isolates to fully-dependent rotatable.

Limitations and interpretative complications.

Hyblock is by no means the procedure of choice for all multivariate causal-path fitting. Although its inductivist competence is importantly beyond the hypothetico-deductive tunnel vision of SEM programs, it conversely lacks the flexible simultaneity of the latter. (By "simultaneity" here I mean solution methods whose errors of model fit are not disposed to cascade, contrasting with sequential fitting procedures such as Hyblock wherein some parameters are finalized before others.) If you use Hyblock to identify the probable dimensionality of your data's factor blocks and which of its direct paths from factors to data variables have near-zero weights, you may well find it subsequently desirable to fit these dimensionalities and zero output paths by a simultaneous structural-modeling solution. Constraints on path coefficients between factors (notably zeros) can also be imposed at that time, with or without heeding Hyblock's solution of [2].

More fundamentally, the limits on what can be learned from analysis of covariances by Hyblock, or indeed any other approach to structural modeling, need unflinching recognition. As a case in point, when Hyblock posits path-dependencies only between factors in selected disjoint blocks, this does *not* implicitly presume further that no causal dependencies obtain within any of these factor blocks. When an appreciable correlation exists between two variables, in particular any pair of latent sources purportedly diagnosed by common factors rotated to oblique simple structure, scarcely ever do we have reason to dismiss the possibility that one is a part-cause of the other unless there is an evident temporal asychrony between them.[8] Second-order factoring of the covariances $\mathbf{C}_{FF}$ diagnosed by a 1st-order factor solution can in principle recover information about causal dependencies among these factors; but the pitfalls to accurate conclusions from such analyses are more treacherous than the extant literature has adequately studied.

Secondly, when Hyblock's rotated solution puts the most salient item loadings in a subset Y_k' of item block Y_k on some factor f_i in a block F_j ($j < k$) path-antecedent to the factors F_k immediate for Y_k, this does not mean that f_i has the same immediacy for the Y_k'-items that it has for items in block Y_j. Rather, it may well be that what is most immediately common to items Y_k' is a factor f_i' that by rights should appear in the F_k block but is too highly correlated with its causal antecedent f_i in F_j to be detectable in the Y_k'-residuals once the block F_j factors are partialled out. Whether a simultaneous SEM refinement of the Hyblock solution can add a term for such an f_i' to factor block F_k and plausibly recover the more articulated path weights from f_i through f_i' to Y_k' I do not know. (I think not, but I could be wrong.)

Finally, Hyblock's difficulty in pulling apart closely correlated factors in separate blocks that by rights our model should distinguish has an interesting manifestation in the creation of non-negligible Waifs. It may well occur that the blockwise-immediate common sources of item totality Y include factors that are singletons relative to the factors that can be recovered just from the covariances within their local item blocks were those to be factored separately, but which are substantially correlated with counterpart factors in other blocks that are likewise locally unique. (For example, if the items are blocked by stages of observation in a longitudinal study, a frugally measured source having at most a single indicator on each occasion may nevertheless be a significant participant in the underlying system dynamics.) If initial factoring is fulsome, as Hyblock advocates, the extraction space received by HYBLOCK may well contain some dimensions which roughly speaking are the major axes of local singletons that form a global cluster. For technical reasons that needn't be detailed here, it is rather unlikely that HYBLOCK can effectively capture such extraction axes by any modest expansion of the subspaces it assigns to those factor blocks. But in HYBLOCK's report on residuals they will be conspicuous as rotated Waifs with large loadings split between blocks, and as explained earlier can be salvaged by HYBALL for study in the final rotated pattern. Even so, we might well prefer these split-block locally singleton Waifs to be modeled as path-structured clusters of global singletons. Conceivably SEM refinement of the Hyblock solution can accomplish this plausibly, albeit I have not been able to envision how.

The larger point to take from these considerations is not so much that Hyblock has inadequacies as an instrument of common-source disclosure--that was certain from the outset--but that there is still much in the theory of source recovery from covariance structures that warrants continued conceptual inquiry and computational development.

An alternative application of Hyblock factor structuring.

If you are squeamish about parsing covariance arrays with prior commitments to their underlying causal-path structure even when the model leaves considerable room for inductive discovery of the structure's details, you can alternatively interpret Hyblock results without causal presuppositions simply as a generalization of rotation to marker variables. (See Overall, 1974, for a precursor of this development.) Classically, a "marker variable" is an empirical indicator whose common part (relative to an adequate array of co-factored variables) is thought to be factorially pure and hence an axis of common-factor space with which it seems appropriate to align one of the rotated factors. Extending this notion, it is easy to see how we might consider the factor subspace spanned by the common parts of a select group of data variables to have sufficient importance that we would like it to be spanned by some subset of our rotated factors, even when factor positioning within that subspace remains negotiable. Put more simply, if we think that the variables in a distinguished subset of our data variables are indicators just of common sources of a special kind, we may want these variables to load in the rotated factor solution just on factors which, precisely because these variables *are* salient on them, we judge to be of that kind. And at bottom, this is all that Hyblock really does.

In brief, Hyblock rotation of factor axes in conformity to a path-structured block model as described earlier is equivalent to choosing one or more not-necessarily-disjoint subsets $\{Y_k^+\}$ of data variables Y as marker groups for repositioning Y's previously extracted common factors in such fashion that for each Y_k^+, the common parts of the variables in marker group Y_k^+ span the same subspace, or nearly so, as a subset F_k^+ of the rotated factors. In this equivalence, following a to-be-explained benign extension of your marker-groups selection, there is a one-one correspondence between marker groups $\{Y_k^+\}$ and the blocks $\{Y_k\}$ of data variables on which HYBLOCK is run, while path relation $\rightarrow$ is now construed first as set-inclusion $\subset$ on $\{Y_k^+\}$, next transferred by isomorphism to an array of disjoint marker blocks $\{Y_k\}$ whose derivation from $\{Y_k^+\}$ is explained below, and finally taken for the path relation imposed by HYBLOCK on the factor blocks $\{F_k\}$ respectively matched with the marker data blocks. In this construction, each marker group Y_k^+ comprises the variables in its corresponding block Y_k together with all variables in the data blocks path-antecedent to Y_k. You will notice that when your marker groups are all disjoint, as would normally be the case when each group is chosen for its presumed factoral purity, set-inclusion is vacuous here and the construction complexities described below become trivial.

Specifically, starting with any list $\{Y_k^+\}$ of marker groups you may desire (though if these groups intersect profusely HYBLOCK's block limit of 30 may be unable to accomodate them), you construct $\{Y_k\}$ and its path structure from $\{Y_k^+\}$ as follows:

A. If your to-be-shifted MODA pattern includes $N_{Y_0} > 0$ manifest inputs, decide whether these are all to be included in every marker group. If they are, ignore them completely until you interpret your final HYBALL-output results. Otherwise, extend your set of Y-variable indices to include $N_Y+1,\ldots,N_Y+N_{Y_0}$ where N_Y is the number of dependent data variables, replace N_Y by $N_Y+N_{Y_0}$ in your notes on how many dependent variables there are, selectively include these additional item indices in whatever marker groups you consider appropriate, and plan to accept HYBLOCK's option of making the fixed-inputs explicit in the block structure.

B. After adding Y (the group of all your data variables) to your initial list $\{Y_k^+\}$ of marker groups, expand this into closure under set-intersection. That is, the expanded marker-group list should have the property that for any two Y_j^+ and Y_k^+ on the list, the variables these have in common are also a group on the list. Then say that Y_j^+ is path-antecedent to Y_k^+ just in case the former is a proper subset of the latter, and re-index the marker groups as needed to embed their path-antecedence in their index order. (There is an auxillary program to work all this out for you.)

C. For each Y_k^+ in your intersection-closed marker-group list, define its core block Y_k to be its subset that includes just the variables in Y_k^+ that are not in any marker group Y_j^+ path-antecedent to Y_k^+. It may turn out that Y_k so constructed is empty; if so, remove Y_k^+ from your marker-group list. (In this case, the Y_k^+-variables' common-parts space is already spanned by the union of axes for the marker spaces path-antecedent to Y_k and hence does not require an additional block.) Then say that core block Y_j is path-antecedent to core block Y_k, with $F_j \rightarrow F_k$ holding on the corresponding to-be-fitted factor blocks, just in case Y_j^+ is path-antecedent to Y_k^+. (Note that path-antecedence on the data groups/blocks, and → on factor blocks corresponding to the core data blocks, is no longer construed as causal influence though allowing that it *might* be causal is not precluded. Rather, it is simply the partial-order relation derived from set-inclusion in the manner just described.)

Steps *B* and *C* set up a block structure from which HYBLOCK can compute a placement of factor axes capturing your initially stipulated marker-group subspaces as wanted. But if that structure is at all complicated, working it out by hand can be quite tedious. Much easier is to run utility program FINDBLK and, when prompted for input, list the marker groups that explicitly interest you. (Each group is entered as a single spaced string of item indices, but the groups' input order is arbitrary.) When the group entries are complete, FINDBLK carries out all of Steps *B* and *C* including proper ordering of the groups/blocks, reports this block structure in an ASCII see-file, and stores it in a transfer file that can be read into HYBLOCK by a single keystroke. From there, you finish just as you would for a causal-path HYBLOCK solution, namely, by first stepping through the blocks to set the factors in each F_k, and then passing this initial block-structured factor positioning to HYBALL for rotation to simple structure under these block constraints. You will probably not have much interpretive use in this case for the factor-block regressions that HYBALL continues to proffer, but you can simply disregard those or instruct HYBALL not to bother.

Both before and after rotation to simple-structure, the factor blocks $\{F_k\}$ set by HYBLOCK have the property that for each block index k, the factors in F_k's union with all factor blocks path-antecedent to it span virtally the same subspace as the common-parts space of marker group Y_k^+. How imperfect "virtual" is here depends on how much variance is left behind when each residual factor block is solved for N_{F_k} principal factors generally fewer than the number N_{Y_k} of variables in block Y_k. When your initial marker-group demands entering Step *A* are fairly modest, that is, contain only a small number of variables, their common-parts subspaces can be captured perfectly by choosing $N_{F_k} = N_{Y_k}$ for all k prior to the last. But if those demands are greedy and require N_{F_k} to be considerably smaller than N_{F_k} for most k, the fit can be rather poor.

REFERENCES

Overall, J. E. (1974). Marker variable factor analysis: A regional principal axes solution. *Multivariate Behavioral Research, 9,* 149-164.

Rozeboom, W. W. (1991a). HYBALL: A method for subspace-constrained oblique factor rotation. *Multivariate Behavioral Research, 26*, 163-177.

Rozeboom, W. W. (1991b). Theory & practice of analytic hyperplane optimization. *Multivariate Behavioral Research, 26*, 179-197.

Rozeboom, W. W. (1992). The glory of suboptimal factor rotation: Why local minima in analytic optimization of simple structure are more blessing than curse. *Multivariate Behavioral Research*, *27*,585-599.

FOOTNOTES

1. This terminology has some unwanted overtones which need to be suppressed: For one, immediate/mediated and direct/indirect are very much relative to the factors made explicit in the model solution. (All paths modeled as direct are surely mediated by variables we have not managed to detect.) And classifying factor-block dependencies as "mediated" vs. "immediate" misleadingly suggests that when $F_i \to F_j \to F_k$, the direct path from F_i to F_k also allowed by our model is ineffectual. (That may be true, of course, but it is not a model presumption.)

2. This presumption is not always reasonable. But for applications where it is not, HYBLOCK allows data variables originally coded as manifest inputs to be reclassified as errorless indicators of 1-dimensional blocks which can be placed in the path structure wherever the user sees fit.

3. Traditionally, we suppress additive constants here by centering all the variables. But these can also be made explicit in [1] by letting one variable in F_0 be the unit dummy and replacing covariances by uncentered 2nd-order moments.

4. In fact, step k+1 is affected by step k only if F_{k+1} is path-dependent on F_k.

5. Saying that two blocks are "path-connected" here means that one is path-antecedent to the other.

6. Since final output from Hyblock includes the rotated factor covariances, it is clearly possible in principle to analyze these for information about the factors' common sources. But what are *good* ways to do this is importantly problematic. For any factor block F_k whose N_{F_k} is large enough to support meaningful factor analysis, the covariances among $\langle F_k, \overset{*}{F}_k \rangle$ can be appropriately analyzed for exogenous common sources G_k of F_k by the Hyball model that takes F_k as dependent measures for which factors $\overset{*}{F}_k$ are fixed inputs whose determination of F_k may be partly or entirely mediated by G_k. But if we try to include larger subarrays of the block-structured $\mathbf{C}_{FF}$ in attempted diagnoses, e.g., of exogenous sources common to several blocks $\{F_k\}$, we find no credible *inductivist* way to do this--which raises questions about how seriously we should take their fit by strongly specified structural equations models.

7. An even deeper question is whether the space spanned by the first N_{F_k} principal axes of residuals E_k is always the best choice for residual-F_k space. But we still lack reasons to prefer any operational alternative. (Note that even were, say, Minres or MLFA clearly superior to principal factoring at common-factor extraction, it would have no special virtue for picking out the most interpretable N_{F_k}-dimensional subspace of the F_k-residuals.)

8. More precisely, when the variables at issue are empirical measures plainly susceptable to contamination by chancy disturbances in the observation procedure, this is true of the variables posited to underlie these measurement outcomes.