parc
Palo Alto Research Center
3333 Coyote Hill Road • Palo Alto • CA • 94304

$ACH_{1.1}$: A Tool for Analyzing Competing Hypotheses
Technical Description for Version 1.1

By PARC AI^3 Team with Richards Heuer
Lance Good, Jeff Shrager, Mark Stefik, Peter Pirolli, & Stuart Card

$ACH_{1.1}$ is an experimental program intended to aid intelligence analysis. It provides a table-oriented workspace for performing the Analysis of Competing Hypotheses (ACH) method (Heuer, 1999). This note provides a technical description of the program, focusing on the scoring algorithms implemented in the tool for relating evidence to hypotheses, known limitations of these algorithms, and brief guidance in their use.

Background: Problems with Intuitive Analysis

Heuer (1999) reviews psychological literature relevant to the performance of intelligence analysis and identifies various cognitive and perceptual limits that impede attainment of best practice. Human working memory has inherent capacity limits and transient storage properties that limit the amount of information that can be simultaneously heeded. Human perception is biased towards interpretation of information into existing schemas and existing expectations. Reasoning is subject to a variety of well-documented heuristics and biases (Tversky & Kahneman, 1974) that deviate from normative rationality. In problem structuring and decision analysis, people typically fail to generate hypotheses, fail to consider the diagnosticity of evidence, and fail to focus on disconfirmation of hypotheses. ACH is designed to ameliorate these problems with intuitive intelligence analysis that arise from human psychology.

The Method of Analysis of Competing Hypotheses

ACH consists of the following steps.

1. Identify possible hypotheses.
2. Make a list of significant evidence for/against.
3. Prepare a Hypothesis versus Evidence matrix.
4. Refine matrix. Delete evidence and arguments that have no diagnosticity.
5. Draw tentative conclusions about relative likelihoods. Try to disprove hypotheses.
6. Analyze sensitivity to critical evidential items.
7. Report conclusions.
8. Identify milestones for future observations.

Figure 1 gives a screen shot of $ACH_{1.1}$, illustrating its table format. The hypotheses under consideration in the example are the columns labeled H1, H2, and H3. Six items of evidence are present in the example in the rows labeled E1 through E6. In the ACH Method, each piece of evidence is assumed to be independent and the hypotheses are exhaustive and mutually exclusive.

As discussed in the tutorial included in the tool, an entry of "I" signals that this evidence is inconsistent with the corresponding hypothesis, and entry of "II" signals that it is very inconsistent with the evidence. The "C" and "CC" entries indicate two levels of consistency.

$ACH_{1.1}$ distinguishes between "I" and "II" in lieu of a detailed representation of how evidence conflicts with a hypothesis. In other words, it models evidence as being contradictory without saying *how* it is contradictory. (A more detailed representation that focuses on causes of contradiction could be useful in generating trees of alternative hypotheses). Rather than employing a symbolic representation of contradiction or a probabilistic one, the ACH method simply provides two levels of inconsistency.

Similarly, $ACH_{1.1}$ provides three levels of weights assigned to evidence. The weight is divided into two general categories of "Credibility" and "Relevance." Roughly, these weights are a stand-in for richer representations of the quality of evidence. Is it reliable? Is it critical to the problem? Is the source authoritative? Or is this "evidence" really just an assumption?

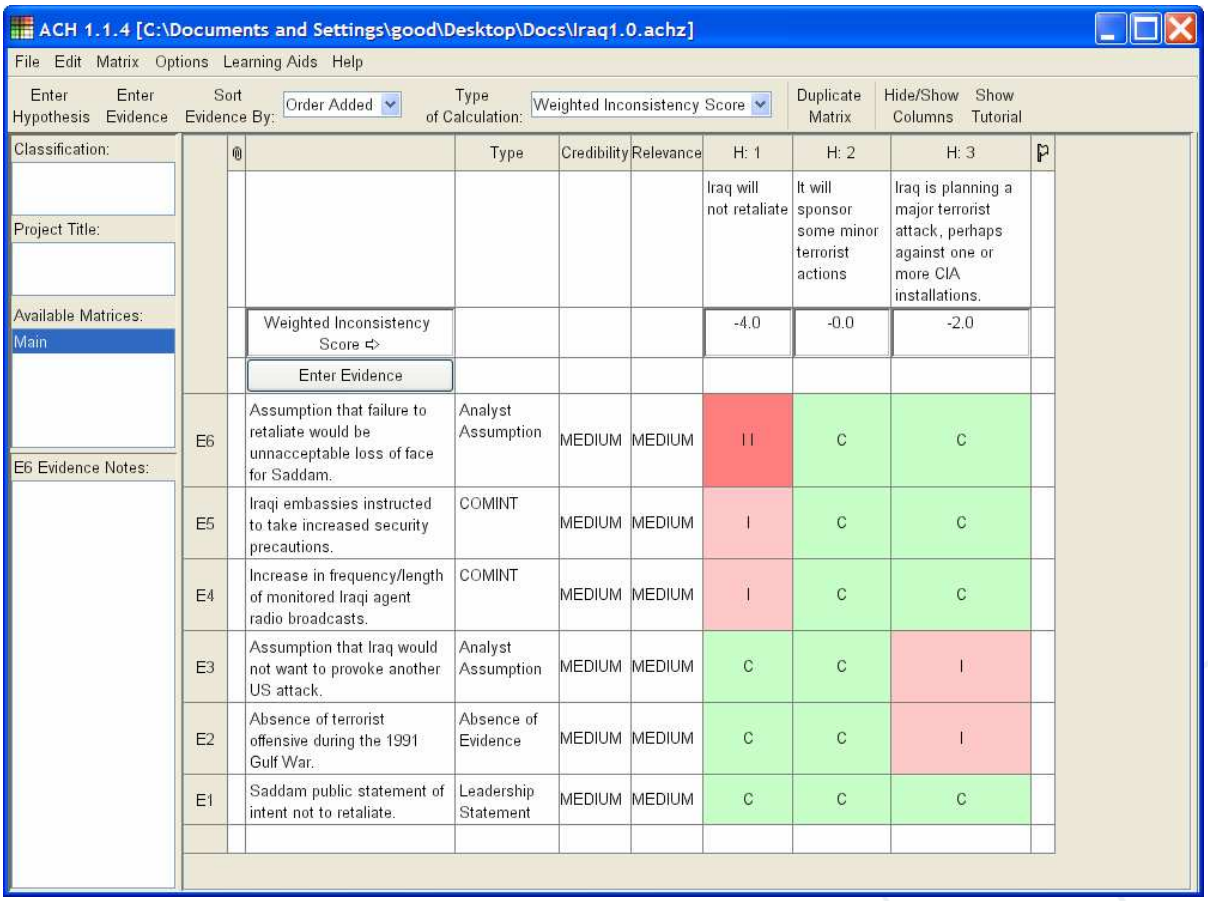

Figure 1. Screen shot of $ACH_{1.1}$.

Broad Caveats for the Method

$ACH_{1.1}$ is intended as a simple tool for organizing thinking about analysis. Its simplicity creates both strengths and weaknesses. Here are some strengths:

- Encourages systematic analysis of multiple competing hypotheses.
- Creates an explicit record of the use of hypotheses and evidence that can be shared, critiqued, and experimented with by others.
- Easy to learn.
- Uses information that analysts can practically understand and enter into the tool.
- Focuses attention on disconfirming evidence – counteracting the common bias of focusing on confirming evidence.
- Does not require precise estimates of probabilities.
- Does not require complex explicit representations of compound hypotheses, time, space, assumptions, or processes.
- Works without a complex computer infrastructure and is available without fee.

Here are some weaknesses.

- **Does not *and cannot* provide detailed and accurate probabilities**.
- Does not provide a basis for marshalling evidence by time, location, or cause.
- Does not provide a basis for accounting for *assumptions*.
- Many of the cognitive steps in analysis are not covered at all. (See Appendix A).

With these caveats $ACH_{1.1}$ can have value when used with a clear understanding of its limitations.

Trade-offs in Accuracy, Practicality, and Understandability

There is a pressing need for accurate and timely intelligence in a world of overwhelming and incomplete data of variable quality. A primary concern of analysis is the principled relating of evidence to hypotheses.

It is desirable that intelligence analysis be accurate, practical, and understandable. It may not be obvious that these criteria can pull in different directions and that it is not always possible to achieve the highest marks in all three at the same time.

Accuracy

Strategic surprise is perhaps the most costly failure of intelligence analysis. Post mortem discussions of such failures are the "bad report cards" of the intelligence community. Several factors contribute to such failures. Among the most general are errors of mindset – where analysis has focused on routine interpretations and overlooked unlikely but high risk ones. Other causes are not generating hypotheses systematically or failing to cross-check analysis with careful and informed review. By accommodating multiple explicit hypotheses and systematic

consideration of available evidence, the ACH method counteracts confirmation bias and some other causes of inaccuracy.

At a more detailed level, accuracy can also mean providing a detailed and accurate analysis of the probabilities of different outcomes.

Practicality

Practicality is important for analytic methods because methods that are impractical are not consistently used. In complex situations, mathematically-sound approaches for estimating and combining probabilities require copious amounts of information in the form of conditional probabilities that are seldom available. In this context, the ACH method takes a position that emphasizes the practicality of working with easily-available kinds of information rather than reaching for the kind of accuracy that might be achievable if much richer models (probabilistic or symbolic) and more information were employed.

Another aspect of practicality is scalability. Does the method continue to work as the size of the problem increases? Other dimensions of practicality include easy-of-use, and requirements on the computational infrastructure.

Understandability

There is a substantial risk in assigning trust to a "black box" whose inner workings are not thoroughly understood. One way of defining understandability is whether a user can give an explanation of what a method does that reasonably predicts both the desiderata used in scoring and the outcome produced. Since users can have different backgrounds (such as math or non-math), what is understandable to one user may not be as understandable to another. Another dimension of understandability is whether the results produced by a method correspond to a user's expectations, based on other kinds of reasoning.

Three Algorithms

These criteria pull in different directions and there are trade-offs in trying to honor them. In the extreme, accuracy can require more data than are practically available, combined by algorithms that are not possible for a person to manually check. At another extreme, algorithms in which it is easy to explain the influence of each new piece of evidence on small problems can be inaccurate and break down on large problems – as the amount of evidence or the number of hypotheses increases.

Given these trade-offs, $ACH_{1.1}$ provides three simple algorithms for scoring evidence: an Inconsistency Counting algorithm, a Weighted Inconsistency Counting algorithm, and a Normalized[1] algorithm. All of these algorithms are intended only as a rough guide for scoring hypotheses. The algorithms operate on the same data, but make different trade-offs.

[1] The Normalized calculation is disabled by default.

An Inconsistency Counting Algorithm

The Inconsistency Counting algorithm is the easiest to explain.

For each item of evidence, a consistency entry of "I" counts -1 against the corresponding hypothesis and an entry of "II" counts -2 against the hypothesis. (All other entries are ignored.) The score for each hypothesis is simply the sum of the counts against it. Restated, the algorithm counts the number of entries that are inconsistent with each hypothesis. The more inconsistent evidence that is entered, the higher the inconsistency score and the less favored the hypothesis.

A Weighted Inconsistency Counting Algorithm

The Weighted Inconsistency Counting algorithm builds on the Inconsistency algorithm but also factors in weights. Suppose that all the entries in the credibility and relevance columns are M (Medium). In this case, the calculation is performed exactly as in the previous non-weighted inconsistency counting algorithm.

When some of the credibility or relevance weights are L (Low) or H (High), the score for each piece of evidence is multiplied by a prescribed value that decreases or increases the influence as intended. In particular, L (Low) is assigned the value 0.707, M (Medium) is assigned the value 1, and H (High) is assigned the value 1.414. This causes high-weighted evidence to have more influence than low-weighted evidence. The Credibility and Relevance weight values are then multiplied together to determine the aggregate weight for a given piece of evidence. The default values used in $ACH_{1.1}$ are as follows:

Credibility	Relevance	I	II
H (High)	H (High)	2	4
M (Medium)	H (High)	1.414	2.828
L (Low)	H (High)	1	2
H (High)	M (Medium)	1.414	2.828
M (Medium)	M (Medium)	1	2
L (Low)	M (Medium)	0.707	1.414
H (High)	L (Low)	1	2
M (Medium)	L (Low)	0.707	1.414
L (Low)	L (Low)	0.5	1

One property of this distribution of weights is that in certain test cases where the weights are systematically changed, the ranked order of hypotheses remains stable. The stability condition is that the ratio (High weight)/(Medium weight) is the same as the ratio (Medium weight)/(Low weight). This condition assures that if all the credibility or relevance weights in an exercise were L or M, and these were systematically changed to M or H in the obvious way, the relative ranking of hypotheses would not change.

From a methodological point of view, the Inconsistency Counting algorithm and the Weighted Inconsistency Counting algorithms implement the main logic of the Analysis of Competing Hypotheses Method in that they focuses attention on disconfirming evidence. They *do not* provide a probabilistic basis for comparing hypotheses.

A Normalized Algorithm

A main virtue of the Normalized algorithm is that *like standard probabilistic models*, the influence of evidence is determined by a multiplicative (or product) approach.

Suppose that all of the evidence items are M (Medium). For each item of evidence, an entry of "I" is assigned a fraction (say .85 corresponding to that probability). These values are analogous to conditional probabilities. (All other entries are ignored.) The raw score for each hypothesis is simply the product of the cell values. The more inconsistent evidence that there is, the lower the score and the less favored the hypothesis. The raw scores for the set of hypotheses are then normalized so that they sum to 1.[2]

As in the Inconsistency Counting algorithm, an entry of "II" is intended to have greater negative influence than an "I" entry. In that algorithm, doubling the influence of a piece of evidence is accomplished by adding together two negative "I" values. In a *multiplication-based* algorithm, to double the influence of a piece of evidence is to multiply the values twice. For an entry of "II" to have twice the influence of "I", the appropriate weight should be .723 (.723 =$.85^2$).

In similar fashion, high-weighted evidence should have more influence than low-weighted evidence. Following the logic of multiplication, if H (High) evidence is intended to have twice the influence of M (Medium) evidence, then the value assigned to H should be the square (second power) of the value assigned to M. Similarly, in order to compute the aggregate weight from the Credibility and Relevance weights the exponent values for both weights are multiplied together. The exponent weights in the normalized calculation currently mirror the weights in the Weighted Inconsistency Score calculation with L (Low) = 0.707, M (Medium) = 1.0, and H (High) = 1.414. This leads to the following table of values in the current version of $ACH_{1.1}$:

Credibility	Relevance	I	II
H (High)	H (High)	$.723 = .85^{2}$	$.522 = (.85^{2})^{2}$
M (Medium)	H (High)	$.795 = .85^{1.414}$	$.632 = (.85^{1.414})^{2}$
L (Low)	H (High)	.85	$.723 = (.85)^{2}$
H (High)	M (Medium)	$.795 = .85^{1.414}$	$.632 = (.85^{1.414})^{2}$
M (Medium)	M (Medium)	.85	$.723 = (.85)^{2}$
L (Low)	M (Medium)	$.891 = .85^{.707}$	$.795 = (.85^{.707})^{2}$

[2] This normalization step reflects the assumption that the hypotheses entered cover all of the possible hypotheses. This assumption is not valid if hypotheses are missing.

H (High)	L (Low)	.85	$.723 = (.85)^2$
M (Medium)	L (Low)	$.891 = .85^{.707}$	$.795 = (.85^{.707})^2$
L (Low)	L (Low)	$.922 = .85^{.5}$	$.85 = (.85^{.5})^2$

A decisive piece of negative evidence, even on a large problem, can have a substantial effect in reducing the score for a hypothesis. For example, a single piece of evidence with a H (High) credibility weight, a H (High) relevance weight, and a "II" entry will reduce the score for the corresponding hypothesis by almost a half (.522). Furthermore, the influence of a piece of evidence does not depend on the order of entry. Nonetheless, the Normalized algorithm is not a true Bayesian model in that it is based on a limited set of subjective probabilities and is limited by its consideration of only disconfirming evidence.

Switching the ACH user interface to use this Normalized algorithm introduces a red box into the top left corner of the table's workspace that is labeled "Potential for Surprise." This is a percentage score that is equal to 100% minus the percentage score of the most likely hypothesis. For example, if the current ACH matrix has 3 hypotheses with scores of 65%, 25%, and 10%, then the "Potential for Surprise" will be 100%-65%=35%. This is designed to give you a rough estimate for the overall uncertainty of the most likely hypothesis.

Summary

$ACH_{1.1}$ is intended as a simple tool that can support the Analysis of Competing Hypotheses method. The ACH method offers benefits for systematically considering multiple hypotheses and avoiding confirmation bias. It is easy to use and provides a basis for documenting the evidence used and the hypotheses considered. It supports a process for generating and comparing hypotheses under circumstances when accurate probabilistic scoring is not feasible.

Nonetheless, the simplicity of the ACH method is not without consequences. Mainly, it neither collects nor incorporates the kinds of information that could be used to create an accurate probabilistic scoring of hypotheses. In creating a computational underpinning for the method, we have developed three algorithms that attempt to provide the usual advantages of a computational substrate without imparting a false sense of precision. The three algorithms make different trade-offs in how they compensate for the lack of a complete probabilistic model.

	Inconsistency Counting and Weighted Inconsistency Counting	Normalized
Calculation	Counts inconsistencies with and without weightings.	Multiplies values corresponding to conditional probabilities for "I"'s.
Provides Normalized Percentage Score	No	Yes
Understandability to non-math users	**Yes & No** (Algorithms simple, but lack percentage-based score for comparisons.)	**No & Yes** (Algorithm appeals better to math types. Provides percentage score.)
Effect of evidence is order-dependent	No – order does not matter.	No
Based on standard probability model	NA	Yes

- The Inconsistency Counting algorithm and Weighted Inconsistency Counting algorithm sidestep having a probabilistic model at all, and simply provides the user with a count reflecting the amount of inconsistent evidence. This approach is the easiest to understand and "tells no lies" but it gives the user perhaps the least intuitive scoring of the hypotheses – assuming that a probabilistic scoring is the most intuitive.
- The Normalized algorithm is the most like a Bayesian approach in that the underlying scoring is based on a multiplicative model. However, like the other algorithms, it is limited in that the ACH method itself does not require the user to enter accurate probabilities. Thus, although the computation seems to behave well at scale, the scores that it computes should not be confused with a more detailed, probabilistic modeling of the evidence.
- All algorithms assume that the pieces of evidence are independent so that their influence on the scoring of hypotheses can be handled independently. Both algorithms approximate reality by classifying evidence with two categories of weights (credibility and relevance) with values of LOW, MEDIUM, and HIGH. This is appropriate for a first cut, but may be insufficiently nuanced for some cases.
- The Normalized algorithm assumes (in its normalization step) that hypotheses are mutually exclusive and exhaustive.
- Users are advised that ACH is at best a guide to thinking. Entering dependent pieces of evidence, leaving out important hypotheses, or entering hypotheses that are not mutually exclusive takes a case outside the simplifying assumptions of the algorithms and could result in "guidance" that is misleading.

This work is funded in part by the Advanced Research and Development Activity NIMD program (MDA904-03-C-0404).

References

Heuer, R. J. (1999). *Psychology of Intelligence Analysis*. Washington, D.C.: Center for the Study of Intelligence.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science, 185*, 1124-1131.

Appendix A –ACH in a Broad Analytic Context

Taken in the large, intelligence analysis is a complex activity involving interlocking processes for gathering information and interpreting it using specialized and often multi-disciplinary expertise.

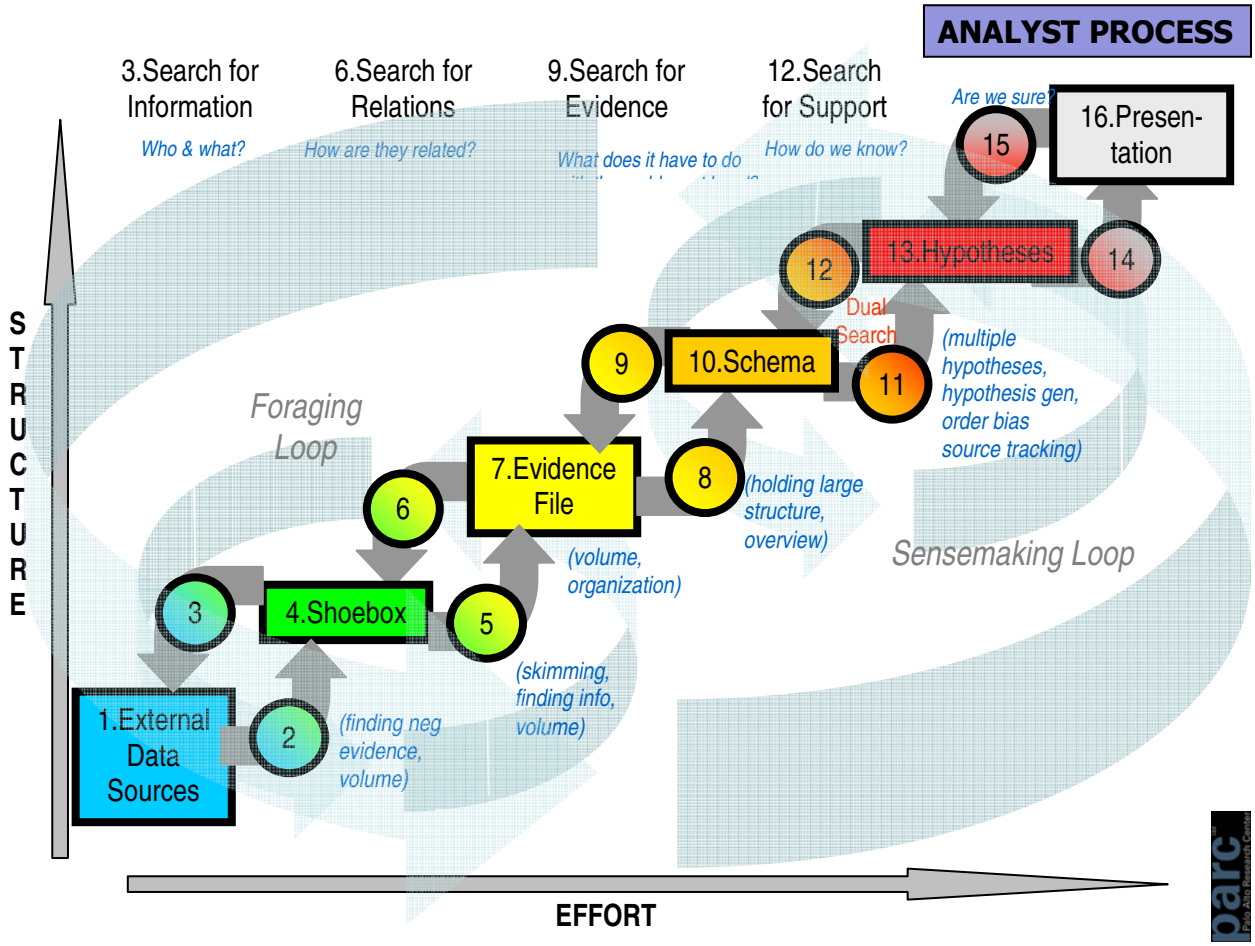

Figure 2. Loops in a Cognitive Model of Analysis.

Figure 2 gives a broad map of steps in a cognitive model of analysis. At the left end of the figure are steps for gathering information and at the other end are steps for managing hypotheses. Although it is tempting at first to read the process sequentially from left-to-right–starting from data collection and ending with hypothesis management and reporting—the loops in the figure represent a much more bi-directional and interlocking process. For example, starting with competing hypotheses, an analyst could request the collection of information that could be used disconfirm some of them. Overall, the figure shows a large "analysis" loop that is subdivided

into two large loops for information foraging loop (steps 1 through 6) and sensemaking (steps 7 through 14). These large sub-loops are further subdivided into more detailed steps.

In such a broad view of analysis, the ACH method (and the $ACH_{1.1}$ tool) are not designed to support the entire intelligence analysis process. The focus of ACH is mainly between steps 7 and 13 – from evidence to hypotheses—without developing explicit reasoning schemas for intermediate reasoning. Computer tools for supporting more detailed reasoning on the intermediate steps would require richer representations of the subject matter and would involve much more detailed reasoning processes. Computer systems that would integrate with processes for gathering and classifying intelligence from information sources or preparing presentations from the analytic work would require substantial integration into the computational substrate.

In summary, the ACH Method and $ACH_{1.1}$ tool focus on a restricted subset of the overall problem. Research on such extensions is being carried out on other projects both as part of the ARDA/NIMD program and at other places.