

Motif discovery by k -TSS grammatical inference *

D. López¹, A. Cano¹, M. Vázquez de Parga¹, B. Calles²,
J.M. Sempere¹, T. Pérez¹, M. Campos¹, J. Ruiz¹ and P. García¹

(1) Departamento de Sistemas Informáticos y Computación. UPV.

(2) Centro Nacional de Biotecnología Universidad Autónoma de Madrid

{dlopez,acano,vmazquez,jsempere,taperez,mcampos,jruiz,pgarcia}@dsic.upv.es
bcalles@cnb.uam.es

Abstract

An important task in biosequences processing is to decide the presence of subsequences with an associated function or specific feature. This task is related to the prediction of the behaviour of the whole protein. In this work, we present a method to determine if a protein contains one of those functional motifs (coiled coil), which is involved in protein interaction. We use a known grammatical inference algorithm to obtain models for both classes involved in the decision task (that is, coiled-coil and non-coiled proteins). The experiments carried out prove the good performance of the approach as it obtains better results than previous approaches.

1 Introduction

The selection of proteins with certain characteristics from genomic sequences is a central goal of computational biology. One aspect of this problem is to detect certain subsequences (known as domains or motifs) with some functional features.

Proteins with *coiled coil* domains are of interest for molecular biologists studying a variety of processes such as protein transport and membrane fusions and the infection of cells by parasites [Skehel and Wiley, 1998][Chan and Kim, 1998]. Predictions based on analysis of primary sequences suggest that approximately 2-3% of all protein residues form coiled coils [Wolf *et al.*, 1997].

The coiled coil is an ubiquitous protein folding and assembly motif made of α -helices wrapping around each other forming a supercoil. The sequences of coiled coils are made of seven-residue repeats $(abcdefg)_n$, called heptads, in which hydrophobic core occurs mostly at positions a and d . The interaction between two α -helices in a coiled coil involves these hydrophobic residues. The result is a highly versatile protein interaction mechanism (see Figure 1). Due to its simplicity and regularity, the coiled coil is the most extensively studied protein motif.

Several programs for predicting coiled coil domains have been described. The most relevant to large-scale annotations

are *coils* [Lupas *et al.*, 1991] (probably the most widely used), *paircoil* [Berger *et al.*, 1995] and *multicoil* [Wolf *et al.*, 1997]. All these programs are based on the probability of appearance of every amino acid in each position of the characteristic heptad, extracted from known coiled coil motifs. Multicoil is the most specialized one, and aims to detect double or triple coiled coil domains.

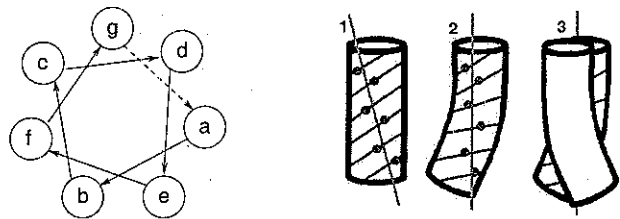


Figure 1: On the left, we show a schematic representation of the relative position of amino acids in a characteristic coiled coil heptad repeat. Residues at the a and d positions are predominantly hydrophobic, whereas those at the e and g positions are frequently charged or polar. Due to the α -helical structure, residues at the a and d position are spatially close one each other. Note that the dashed connection links the g -site of one heptad with the a -site of the following one. On the right, we schematically show how the hydrophobic channels of two coiled coil motifs (1 and 2) can interact (3). Notice that the bullets represent those amino acids in sites a and d .

Lupas *et al.* [Lupas *et al.*, 1991] take into account that even very short proteins have stable coiled coils containing four or five heptads. A protein sequence is analyzed using a sliding window of 28 amino acids. A score for each amino acid in the sequence of the protein is obtained using the probabilities of appearance for each amino acid in protein sequences. The score distributions for general globular proteins and coiled coil sequences are approximated with Gaussian curves (namely G_{cc} for coiled proteins and G_g for general globular ones). The processing of an unknown protein starts by obtaining the scores for each amino acid of its sequence. For any given score S , the probability that such score belongs to a coiled coil motif is obtained according to the following function:

*This work is partially supported by Spanish CICYT under contract TIC2003-09319-C03-02. The support of IJCAI, Inc. is also acknowledged.

$$P(S) = \frac{\alpha G_{cc}(S)}{(\beta G_g(S) + \alpha G_{cc}(S))}$$

Thus, for each amino acid of the protein, a probability of belonging to a coiled coil motif is obtained. The authors approximate the values of α and β , setting them to 1 and 30 respectively.

Although this approach is widely known by the biological community, it is known that the method leads to a significant number of *false positives*, some of them due to the continuous appearance of some frequent amino acids in coiled coil regions (i.e. $(Lys - Lys - Lys)_n$ scores highly though it is not a coiled coil). To solve this problem, Berger et al. [Berger et al., 1995] propose a method that takes into account the pairwise amino acid correlations in known coiled coils. The authors obtain a score for each amino acid following the general scheme from [Lupas et al., 1991], but they consider correlations between amino acids where Lupas considers probabilities of appearance. The correlations and the size of the window used were empirically selected, thus,

- the correlations between the pairs of amino acids placed in positions $(i, i + 1)$ and $(i, i + 4)$ were considered
- the size of the sliding window was set to 30.

The authors claim that the approach is useful to discard false positives detected by the Lupas' approach. They carry out a wide experimentation to show the behaviour and present several examples of false positives detected. This approach is also widely known and used by the scientific community.

Nevertheless, the problem of locating general coiled coil motifs is far of being solved. Several authors have noted several important coiled-proteins that are not detected when the previous approaches are used (among others, fusion-membrane proteins of the human and simian immunodeficiency virus or Ebola virus [Singh et al., 1999]). Thus, several other works propose solutions for more specific instances of the problem [Singh et al., 1998][Singh et al., 1999].

In our work, we use a grammatical approach to decide whether a protein contains or not a coiled coil motif. This approach to the processing of biosequences has been used previously by Yokomori et al. [Yokomori et al., 1994]. In his work Yokomori uses inference of k -TSS languages in a protein α -chains identification task. The experimentation carried out by Yokomori uses a low number of samples for training the models (20 samples out of the 123 positive and 2567 negative samples available), with no error correcting analysis to analyze the test set. Nevertheless, the results show good performance of the method. López et al. in the detection of coiled coil domains. They are limited due to the lack of a parser able to deal with large automata and sequences [Lopez et al., 2004], nevertheless their results are encouraging.

We tackle the problem of determining if a given protein contains at least a coiled coil motif. We will also use grammatical inference algorithms in order to obtain models for the classes involved in the problem, but we use the whole sequence of the protein instead using the subsequences that contain such motifs (such approach was used by Yokomori in his work). The experimental results obtained show that the performance of our method is comparable to those described

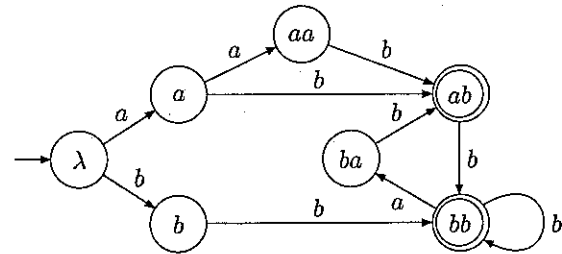


Figure 2: Automaton obtained for the training sample by the k -TSS inference algorithm with $k = 3$

above. This work is structured as follows: in section 2 we introduce the notation and the inference algorithms that will be used. In section 3 the experimental results are exposed and discussed. The conclusions of the work and some lines of future work end this paper.

2 Detection of functional motifs in biosequences by grammatical inference

We will next describe some theoretical aspects that will be used through the paper.

Let I_k, F_k be two sets of strings of length less than or equal to $k - 1$, and S_k a set of strings of length k . A given language L is said to be k -testable in the strict sense (k -TSS) if the words in L start by a prefix of I_k , end by a suffix of F_k and do not contain any segment of S_k .

The family of k -TSS has been deeply studied and several inference algorithms have been proposed [García et al., 1990][García and Vidal, 1990]. In these works, the authors prove that the proposed inference algorithm obtains the minimal automaton which accepts the smallest k -TSS language that contains the training set. The inference algorithm for k -TSS languages has been successfully used in several pattern recognition tasks [Bordel et al., 1994][Cruz and Vidal, 1998][Torres and Varona, 2001].

In order to show the behaviour of the algorithms, let us consider the following training set: $M = \{aabb, abbb, abba, bbb\}$, in Figure 2 we show the automaton obtained by the k -TSS inference algorithm for $k = 2$.

In this work, we used the KTSS algorithm to infer both the class of proteins that contains a coiled coil motif and the class of proteins that does not contain such motifs. Once the classes were obtained, we used the Viterbi algorithm (i.e. [Amengual et al., 2001]) to obtain the probability that a protein sequence belongs to every model. The nature of the problem leads us to deal with large automata and very long sequences (greater than 3000 symbols in some cases). Our implementation of Viterbi's algorithm does not consider insertion/deletion operations (those operations has been considered, but do not offer better results), and allows us to carry out the analysis of the sequences in a standard personal computer without extra requirements.

All the protein sequences were extracted from SwissProt (release 40, April 2003). All the entries in the database are annotated with the known domains (motifs). Furthermore,

```

DR PROSITE; PS00226; IF; 1.
KW Intermediate filament; Coiled coil.
FT DOMAIN 1 84 HEAD.
FT DOMAIN 85 433 ROD.
FT DOMAIN 434 589 TAIL.
FT DOMAIN 85 116 COIL 1A.
FT DOMAIN 117 130 LINKER 1.
FT DOMAIN 131 268 COIL 1B.
FT DOMAIN 269 285 LINKER 12.
FT DOMAIN 286 433 COIL 2.
SQ SEQUENCE 589 AA; 67694 MW; 5EF7D...
SLKQSQESSE YEIAYRSTIQ PRTAVRTQSR QSG...

```

Figure 3: Fragment of a example protein from the database. Note that the protein contains three subsequences corresponding to coiled coil motifs

some potential domains (those whose function has not been yet assured) are also annotated. Figure 3 is one example entry of the database.

We extracted from the database the sequences of those proteins with non-potential annotated coiled coil domains. The resulting set of 350 sequences will be referred to from now on as M_c in the sequel. We also randomly extracted a bigger set of 3500 sequences from de database. All the sequences in this set correspond to proteins without references to coiled coil domains. We will refer to this set as M_{nc} . Note that the absence of annotations for a domain does not mean that the protein does not contain it, given that it is possible that the protein has not been studied in that way.

3 Experimentation and results

The resulting sequences could be seen as strings in an alphabet over 22 symbols (20 amino acids plus the glutamic and aspartic acids¹). To reduce the size of the alphabet as much as possible without loss of information, two different codifications were considered. The first one considered the hydrophylic properties of the amino acids, classifying them in two classes: hydrophobic and polar [Clote and Backofen, 2000]. The second one is due to Dayhoff and has been used previously in a related work (identification of protein α -chains [Yokomori *et al.*, 1994]). This codification is based on some physical-chemical properties of the aminoacids (acidity, aromaticity, hydrophobicity, among others). Although the original code considers six classes (from *a* to *f*) we have extended it to consider the glutamic and aspartic acids as a new class. Figure 4 shows the correspondence of each amino acid for both codifications.

The same process is applied to each codification of the database. We extracted a training set from M_c and M_{nc} and inferred an automata using those sets (namely TR_c and TR_{nc}) and both algorithms exposed above. Using the same set of samples, probabilities were assigned to the transitions of the automata.

¹It is possible to find sequences that contains the symbol *X*. This symbol appears when it is not clear which amino acids occupy a certain position. In this work, we do not consider such sequences.

| amino acid | P/H | Dayhoff |
|------------|-----|---------|
| C | p | a |
| R, H, K, | p | d |
| D, E | p | c |
| N, Q | p | c |
| B, Z | p | g |
| Y | p | f |
| G | p | b |
| S, T | p | b |
| A, P | h | b |
| F, W | h | f |
| L, V, M, I | h | e |

Figure 4: Amino acid codifications.

To obtain an error model, that is, the probability to edit each symbol in an error-correcting analysis of the test sequences, two disjoint sets of samples respect TR_c and TR_{nc} , namely TB_c and TB_{nc} , were extracted from the datasets. This table was constructed using Viterbi's algorithm.

Finally, test sets (TS_c and TS_{nc}) were extracted from the database, and membership probabilities were obtained by using Viterbi's algorithm. The proteins were classified using a maximum probability criterion.

Each experiment involved the set M_c and one subset of M_{nc} of 350 samples. In order to obtain as much statistical relevance of the results as possible, seven different balanced partitions of the data were considered. The influence of the codification and the parameter k were also studied.

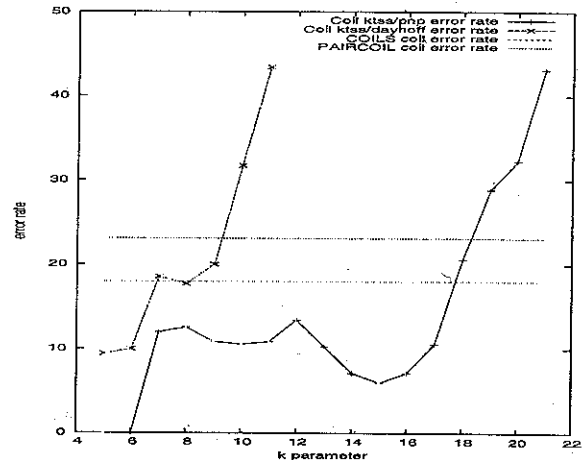


Figure 5: Error rate obtained when only coiled-coil sequences were considered

Figures 5,6 and 7 show the results obtained. Note that the richer the codification used the lower the value of the parameter needed to obtain good results. It has to be noted that Dayhoff's codification leads to improve the false positive error rate obtained by PAIRCOIL while it maintains high detection rate. As mentioned before, this method is characterized by

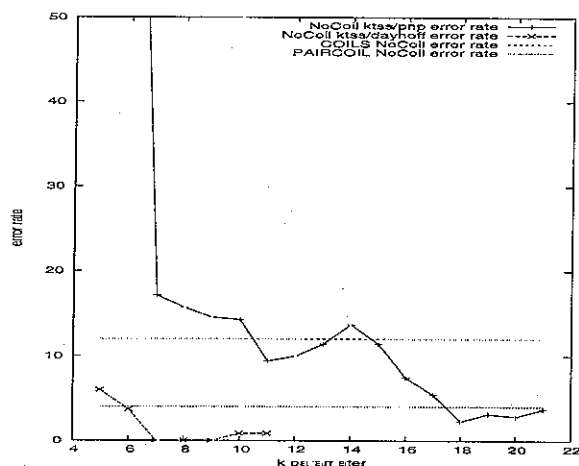


Figure 6: Error rate obtained when non-coiled sequences were tested.

the low number of false positives that provides, obtaining this feature by reducing the detection rate. The binary codification needs higher values of the parameter, leading to similar error rate when coiled and non-coiled sequences are considered independently.

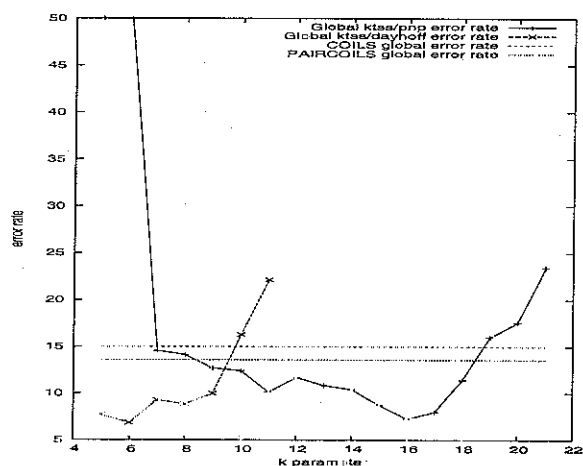


Figure 7: Global error rate obtained by the different approaches.

We compared our results with the most known prediction algorithms [Lupas *et al.*, 1991][Berger *et al.*, 1995] using them somewhat in *detection mode*, that is, we run available versions of the algorithms ([NCOILS][PAIRCOIL] respectively). As stated before, these algorithms give for each amino acid a probability to belong to a coiled coil domain. To compare those methods with ours, we set the threshold to the default value (0.5) and considered as a coiled-coil protein those sequences containing at least one symbol that reaches such threshold.

| | | M_c | M_{nc} | Global | |
|------------|----------|---------|----------|--------|-------|
| error rate | Coils | 0.180 | 0.120 | 0.150 | |
| | Paircoil | 0.231 | 0.040 | 0.136 | |
| | ECGI | 0.155 | 0.378 | 0.267 | |
| | k -TSS | PNP | 0.071 | 0.074 | 0.073 |
| | | Dayhoff | 0.100 | 0.037 | 0.068 |

Table 1: Comparison between algorithms. The k -TSS automata were inferred considering the binary and Dayhoff's codification. 200 samples were used for training and 100 samples for obtaining the error table. The table shows the results for $k = 16$ and binary codification and $k = 6$ and Dayhoff's codification.

Table 1 show the global behaviour of every method tested when applied to the sets M_c and M_{nc} . Notice that the learning of k -TSS languages lead to best classification of the sets independently of the codification. Binary codification lead to balanced error rates, while Dayhoff's codification allow to obtain similar results with lower number of false positives. ECGI results were extracted from [Lopez *et al.*, 2004] and show worse behaviour.

4 Conclusions

This work proposes a new method to detect the presence of a coiled coil motif within a protein. Coiled coil motifs are frequently involved in protein-to-protein interactions, and play central roles in diverse processes such as cell-invasion (i.e. HIV, SIV [Marti *et al.*, 2004] and Ebola virus [Watanabe *et al.*, 2000]), protein trafficking, signalling and transcription. It is also one of the principal subunit oligomerization motif in proteins.

A grammatical inference approach was used to detect such motifs in the protein sequences. Thus, we considered the inference of k -TSS languages and compared the results with previous work. First of all, we inferred automata for both classes of coiled and non-coiled proteins. The test set was analyzed using the Viterbi algorithm and classified with a maximum probability criterion. The experimental results show that the approach obtains better global results, no matter which codification was used. Taking into account the differences on the behaviour of previous methods, Lupas' method (coils) has higher detection rate than Berger's method (paircoil). This is the main reason that makes coils to have higher false positive rate than paircoil. Considering the results of our method using the binary codification, the shape of the results are similar to those obtained by coils. When Dayhoff's codification is used, the behaviour changes and the approach lead to lower false positive rates.

During the experimentation we notice abnormal error models for high values of the parameter. *Normal* error models give more probability to non edit operations, but the error models obtained did not follow this behaviour for some symbols. The nature of the task lead to deal with large automata and long sequences. This fact and the lack of a bigger dataset could be the main reasons for this behaviour. It remains to consider standard error models built by hand, which should be done in future work.

It is very important to note that the database contains annotations only for those proteins that contain a coiled coil region. Several entries in the database contain subsequences annotated as *potential* coiled coil, but those annotations are only based on sequence homology (alignment) with other coiled coil regions. Therefore, it is not possible to assure that those subsequences correspond to truly coiled coil regions. Besides, the database does not contain any annotation to assure that a protein does not contain a coiled coil region. Furthermore, as discussed above, there exist several algorithms devoted to detect coiled coil domains in specific families of proteins [Singh *et al.*, 1999][Singh *et al.*, 1998]. This leads us to think that some other protein families contain coiled sequences undetected by traditional approaches.

This fact lead us to think that the results should be easily improved by considering a wholly annotated database. This could be done by considering a structural database, where tridimensional information of the proteins is stored. This information should be used to filter those potential sequences. This way a more confident database could be obtained.

The method proposed by Berger *et al.* [Berger *et al.*, 1995] is useful to discard false positive coiled coils proteins, and it is usually a second step after applying the method by Lupas *et al.* It should be studied if a similar two-step approach, using our method, would obtain better results.

Coiled coil are well characterized motifs and its structure is the key stone of the most used prediction algorithms [Lupas *et al.*, 1991][Berger *et al.*, 1995]. Due to its regularity, the coiled coil motif has been extensively studied. The present study permits us to conjecture that some other important motifs, whose structure is poorly known, could be detected by using grammatical inference techniques.

References

- [Amengual *et al.*, 2001] J.C. Amengual, A. Sanchis, E. Vidal, and J. Benedí. Language simplification through error-correcting and grammatical inference techniques. *Machine Learning*, 44:143–159, 2001.
- [Berger *et al.*, 1995] B. Berger, D.B. Wilson, E. Wolf, T. Tonchev, M. Milla, and P. S. Kim. Predicting coiled coils by use of pairwise residue correlation. *Proc. Natl. Acad. Sci.*, 92:8259–8263, 1995.
- [Bordel *et al.*, 1994] G. Bordel, I. Torres, and E. Vidal. Back-off smoothing in a syntactic approach to language modelling. In *Proc. of the 1994 International Conference on Speech and Language Processing*, pages 851–854. IC-SLP, 1994.
- [Chan and Kim, 1998] D.C. Chan and P.S. Kim. Hiv entry and its inhibition. *Cell*, 93:681–684, 1998.
- [Clote and Backofen, 2000] P. Clote and R. Backofen. *Computational Molecular Biology*. John Wiley and Sons Ltd., 2000.
- [Cruz and Vidal, 1998] P. Cruz and E. Vidal. Learning Regular Grammars to Model Musical Style. *LNAI*, 1433:211–222, 1998. 4th International Colloquium ICGI-98.
- [García *et al.*, 1990] P. García, E. Vidal, and J. Oncina. Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, pages 325–328, 1990.
- [García and Vidal, 1990] P. García and E. Vidal. Inference of k -testable Languages in the Strict Sense and application to Syntactic Pattern Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 920–925, 1990.
- [Lopez *et al.*, 2004] D. Lopez, A. Cano, M. Vazquez de Parga, B. Calles, J.M. Sempere, T. Perez, J. Ruiz, and P. Garcia. Detection of functional motifs in biosequences: A grammatical inference approach. In X. Messeguer and G. Valiente, editors, *Proceedings of the 5th Annual Spanish Bioinformatics Conference*, pages 72–75. Univ. Politècnica de Catalunya. ISBN: 84-7653-863-4, 2004.
- [Lupas *et al.*, 1991] A. Lupas, M. Van Dyke, and J. Stock. Predicting coiled coil from protein sequences. *Science*, 252:1162–1164, 1991.
- [Marti *et al.*, 2004] D.N. Marti, S. Bjelić, M. Lu, H.R. Bosshard, and I. Jelesarov. Fast folding of the hiv-1 and siv gp41 six-helix bundles. *J. Mol. Biol.*, 2004.
- [NCOILS] Source Code NCOILS. <http://www.russell.embl.de/cgi-bin/coils-svr.pl>.
- [Newman *et al.*, 2000] J.R. Newman, E. Wolf, and P.S. Kim. A computationally directed screen identifying interacting coiled coils from *saccharomyces cerevisiae*. *Proc. Natl. Acad. Sci.*, 97:13203–13208, 2000.
- [PAIRCOIL] PAIRCOIL implementation by the authors. Available at: <http://theory.lcs.mit.edu/bab/computing>.
- [Singh *et al.*, 1998] M. Singh, B. Berger, P.S. Kim, J.M. Berger, and A.G. Cochran. Computational learning reveals coiled coil-like motifs in histidine kinase linker domains. *Proc. Natl. Acad. Sci.*, 95:2738–2743, 1998.
- [Singh *et al.*, 1999] M. Singh, B. Berger, and P.S. Kim. Learncoil-vmf: Computational evidence for coiled-coil-like motifs in many viral membrane fusion proteins. *J. Mol. Biol.*, 290:1031–1041, 1999.
- [Skehel and Wiley, 1998] J.J. Skehel and D.C. Wiley. Coiled coils in both intracellular vesicle and viral membrane fusion. *Cell*, 95:871–874, 1998.
- [Torres and Varona, 2001] I. Torres and A. Varona. k -ISS language models in speech recognition. *Computational Speech and Language*, 15:127–149, 2001.
- [Watanabe *et al.*, 2000] S. Watanabe, A. Takada, T. Watanabe, H. Ito, H. Kida, and Y. Kawaoka. Functional importance of the coiled-coil of the ebola virus glycoprotein. *Journal of Virology*, 74(21):10194–10201, 2000.
- [Wolf *et al.*, 1997] E. Wolf, P.S. Kim, and B. Berger. Multicoil: a program for predicting two- and three-stranded coiled coils. *Protein Science*, 6:1179–1189, 1997.
- [Yokomori *et al.*, 1994] I. Yokomori, N. Ishida, and S. Kobayashi. Learning local languages and its application to protein α -chain identification. In *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pages 113–122. IEEE, 1994.