Pathway Representation using Finite State Automata and Comparison using the NCI Thesaurus

By

Samuel Leung
B.Sc., University of British Columbia, 2000

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Supervisory Committee:

Dr. B. Kapron, (Department of Computer Science)

Dr. V. King, (Department of Computer Science)

Dr. A. Thomo, (Department of Computer Science)

Dr. M. Wilkinson (Department of Medical Genetics, UBC)

Supervisory Committee:

Dr. B. Kapron, Supervisor (Department of Computer Science)

Dr. V. King, Departmental Member (Department of Computer Science)

Dr. A. Thomo, Outside Member (Department of Computer Science)

Dr. M. Wilkinson, External Examiner (Department of Medical Genetics, UBC)

ABSTRACT

Can one classify biochemical pathways based on their topology? What is the topology of a biochemical pathway? What are the fundamental principles underlying different biochemical pathways involved in similar functional areas? Will one be able to characterize pathway "motifs" similar to motifs in proteins – i.e. reoccurring patterns in pathways? This thesis describes an attempt to develop a quantitative framework for the general representation and comparison of biochemical pathways. This quantitative framework involves a mathematical model to represent biochemical pathways and a set of similarity criteria to compare these biochemical pathways. We anticipate that such a tool would allow biologists to answer important questions such as the ones mentioned above.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I thank God for His love and grace.

I thank my mother and father for their sacrificial love and support.

I thank my wife, Vivian for her love, kindness and patient endurance.

I thank my supervisor, Dr. Bruce Kapron for his guidance and support.

Dedications

In loving memory of my mother, Ada.

Epigraph

"In the beginning God created the heavens and the earth."

~ Genesis 1:1

Preliminaries

Biology Preliminaries

For the background reading of general biochemistry, please refer to the following textbooks: [1], [24], and [32]. The following is some background information on biochemistry to aid the reader in understanding the material in this thesis.

Major Types of Macromolecules

Protein

Protein is made up of one or more chains (polymers) of amino acids. There are twenty types of amino acid molecules that are common to all organisms. Almost all naturally synthesized proteins in all organisms are composed of a combination of these twenty types of amino acids. These different types of amino acids are often represented in the literature as letters, for example: K, V, Y, T and H. A chain of amino acids is often written as a sequence of letters in the literature. The function of a protein depends largely on its 3-dimensional structure and the chemical properties of certain amino acids at certain areas of the protein. The structure of a protein can be determined using X-ray crystallography or nuclear magnetic resonance (NMR). In order to do these experiments, one would need to obtain a decent quantity of purified protein in its native conformation, and this is often very difficult. Determining the amino acid sequence of the proteins is often much easier given today's advances in biotechnology. For example, short amino acid sequences can be determined using Edman degradation (a series of chemical reactions) performed by automated machines. Protein sequences can be determined by

mass spectrometry via matching between peptide fragments and mass signatures of peptides with known amino acid sequences. One would therefore, hope to be able to predict how the chain of amino acids folds into a functional protein given its amino acid sequence (the primary structure of the protein) – the 3-dimensional structure of a protein depends largely on its amino acid sequence. Currently, the most effective way of determining the 3-dimensional structure of a protein is to compare (align) an unknown amino acid sequence with amino acid sequences of known 3-dimensional structures (e.g. as determined by X-ray crystallography). This is one of the reasons why sequence alignment is studied very intensely.

Nucleic Acid

Nucleic acid includes deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Most nucleic acids function as a genetic information transfer medium. Most prominent is the genomic DNA, which stores the necessary information to "build" the organism. There are four types of DNA and RNA molecules that are involved in genetic information processing. They are often represented as letters: A, T, C, and G for DNA, and A, U, C, and G for RNA. Nucleic acids are involved in other roles as well. For example, adenosine triphosphate (ATP) functions as an immediate energy source for the cell, ribosomes (consisting of RNA) are involved in protein synthesis, and guanine triphosphate (GTP) is involved in many signal transduction pathways.

The central dogma of cell biology illustrates the significance of the involvement of DNA and RNA in genetic information processing. The "central dogma" describes how proteins are synthesized from the information stored in genomic DNA. The genomic DNA (e.g. chromosome in human) stores the protein's amino acid code. In a process

called transcription, the cell generates a copy of a portion of the DNA that codes for a particular protein in a form of RNA – messenger RNA (mRNA). The mRNA, which is a chain of RNA molecules, is then translated, in a process called translation, into a chain of amino acids. The chain of amino acid is then folded into a functional protein. In addition to coding for protein, genomic DNA also contains regulatory sections or units. These regulatory units determine how and when transcription should occur. The most popular approach in predicting the function of DNA is to compare (align) DNA of unknown functions to DNA's with known function. Methods in sequence alignment, as applied to nucleic and amino acid sequences, have been researched very actively for over half a century [9].

Carbohydrates and Lipids

These are two other types of macromolecules. They play important roles in cell biology. However, understanding of these macromolecules is not required for the materials discussed in this thesis.

Biochemical Pathways

There are a number of different definitions for biochemical pathway. One definition of pathway that is used often in bioinformatics is as follows: a sequence or network of genes that are transcribed in a causal manner. For example, if the pathway is represented as a graph, the nodes would represent genes and the edges would represent "cause this gene to express" or "turn on this gene" relationships.

Another definition, the one that is used in this thesis, is as follows: a sequence of chemical reactions that together achieve a purpose. What exactly is considered a

pathway depends on the definition of the purpose of a pathway. If the purpose is of a larger scope, the pathway might include more chemical reactions or other smaller pathways or even cells interacting with each other, as in the case of immune response. If the scope is smaller, fewer reactions will be included.

Therefore, the term "pathway" is used among biologists to refer to biological processes in differing perspectives and/or levels of detail. In the literature, it should be clear from the context which definition of "pathway" is used.

The following are two examples of biochemical pathways.

Apoptosis (Cell Death) via CD95

This pathway is involved in a certain type of cell death, apoptosis, where a cell commits suicide and clean up after itself. This is in contrast to necrosis where the cell, after it commits suicide, can cause inflammation. This pathway is an example of a signal transduction pathway where a biological signal, which asks the cell to commit suicide, is transmitted from the cell membrane to some molecular machinery inside the cell via some chemical reactions. The reactions involved include the following. The intercellular biological signal, a cytokine (CD95 ligand), binds to and activates the cytokine receptor (CD95) in the cell membrane. CD95 than activates an adaptor protein, FADD, which activates an enzyme, caspase-8, which activates another enzyme, caspase-3, which activates other enzymes to break down the cell and to package the toxic remains of the cell.

Glycolysis

This is an example of metabolic pathways – pathways involved in breaking or building molecules. In this case, glycolysis breaks down the sugar molecule (sucrose) to extract energy from it. The end product of this pathway, acetyl-CoA, can be used in another pathway, Citric Acid Cycle, for further extraction of energy. This pathway involves a series of enzymatic reactions. A typical enzymatic reaction can be described as follows. First, one or more reactants bind to an enzyme. This causes an enzyme to undergo a conformational change (i.e. changes its shape). This will assist the reactants, bound to the enzyme, to undergo one or more chemical reactions. When the reaction(s) has finished, the enzyme will release the products of the reactions. An enzymatic reaction is often depicted in the textbooks in the following way.

$$REACTANT(s) \xrightarrow{ENZYME} PRODUCT(s)$$

The following is a diagram of glycolysis from http://www.biocarta.com. Please refer to Appendix C. "BioCarta Diagram Legend" for the legend of this figure.

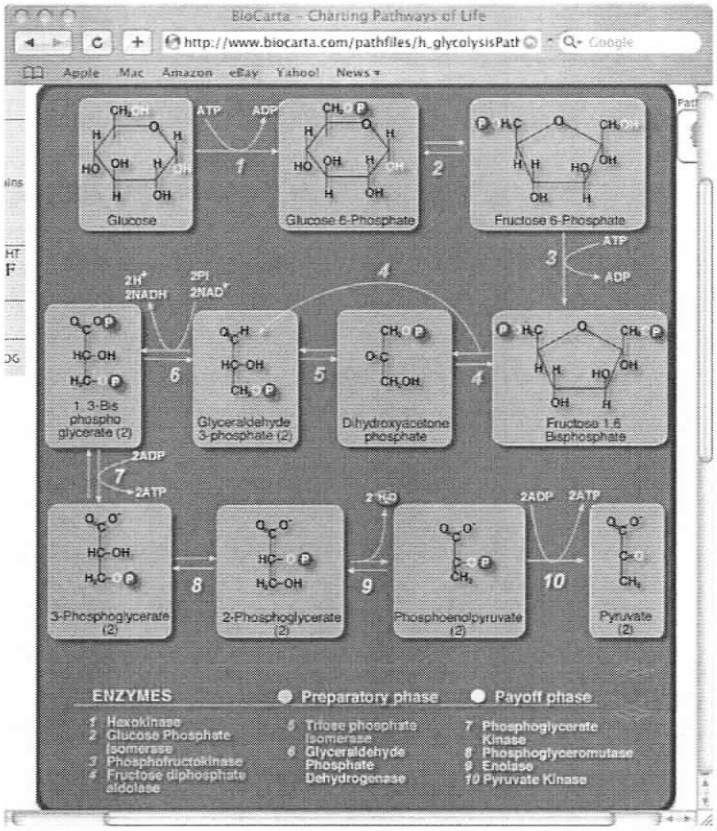


Figure 0-1: Diagram of glycolysis in www.biocarta.com.

Pathway Database

Data on biochemical pathways have been accumulated for over a century, and recent advances in molecular biology have enable scientists to produce an explosive amount of pathway data. As a result, databases on biochemical pathways are now available. Pathway databases come in different flavors, some may contain extensive information on the genes and structure of the chemical agents that are involved in the pathway, while others may contain only links to other public databases. Some are intended to be used mainly through a web browser, while others provide database access through application interface (API), which would allow a computer program to access the data in the database. The following are some examples of popular public pathway databases. For a more comprehensive guide (catalogue) of databases in biology, Nucleic Acids Research,

a very popular journal in the biology community, has a compilation of biology databases that appeared in this journal [14]. The reader can refer to this resource at the following web page: http://www3.oup.co.uk/nar/database/.

Biocyc [55]

This is a suite of pathway/genome databases from Stanford Research Institute (SRI) International [39][40]. The web interface can be accessed on the following site: http://www.biocyc.org. Each database contains information on the genome and metabolic pathways of a single organism with the exception of one database (MetaCyc), which contains metabolic pathways information from many organisms. Currently, BioCyc provides databases on 14 species including Human and E. Coli. MetaCyc contains information from 160 species. The only way to access these databases is through the web interface. One could also download these databases and analyze them using their pathway analysis tools, "Pathway Tools" (http://bioinformatics.ai.sri.com/ptools/). The analysis that the user can perform with these tools includes creation, editing, querying, and visualization.

KEGG [59]

Kyoto Encyclopedia of Genes and Genomes (KEGG) [26] is a project in the Kanehisa Laboratory of Kyoto University of Bioinformatics. The web interface, for academic users, can be accessed on the following site: http://www.kegg.org. KEGG contains information on metabolism, genetic information processing, environmental information processing, cellular processes, and human diseases. Unlike BioCyc, KEGG does not group the pathways by organisms. Instead, for each individual reference pathway, the

user can select the specific pathway, as they exist in various organisms[1]. In addition to the web interface, the user can access KEGG through their application interface (API). The API server is accessed through SOAP (Simple Object Access Protocol) over HTTP (Hypertext Transfer Protocol). KEGG provides API libraries for the following programming languages: Perl, Ruby, Python, and Java. An example of an API query (in Java) would be:

```
public String[] get_compounds_by_pathway(String pathway_id)
```

STKE [63]

Signal Transduction Knowledge Environment (STKE) contains information on the current findings in signal transduction pathways. STKE is essentially a knowledge base, a resourceful web page with links to other databases/websites, rather than a database, which one would expect to be able to construct queries. STKE does not support any queries as found in BioCyc and KEGG (i.e. find all genes in pathway xxx). One can only browse through the database using their predefined categories. For example, the categories for browsing by pathways include "Subject", "Scope", "Model Organism", and "Science[2] Issue". Unlike BioCyc and KEGG, which may contain computationally generated pathways, all pathways in STKE are derived from literature. Pathway diagrams are known as "Connection Map" in STKE. One can access STKE through http://www.stke.org.

---

[1] Many organisms use the "same" pathways. For example, many organisms such from E. Coli to Human, use glycolysis. However, the players in these pathways (e.g. enzyme), though they have the same function, can be quite different (e.g. their amino acid sequences can be different).

[2] This refers to the scientific magazine, *Science*.

BioCarta [54]

Similar to STKE, BioCarta (http://www.biocarta.com) is a knowledge base rather than a database. One could browse through this database via predefined categories. All pathways are derived from literature. BioCarta is a more general pathway database comparing to STKE as it contains more pathways such as those involved in metabolism.

Computer Science Preliminaries/Conventions

Automata Theory

Please refer to [23] for an introduction to automata theory. Automata theory is a mathematical model used to model the dynamic behavior of a system based on inputs and transitions. It has been used to model behaviors of software and hardware systems and it is often used to proof robustness of protocols. An automaton (or state machine) consists of the following:

- A set of states, Q.
- A set of input symbols, Σ.
- A state transition table – a mapping between current state, input symbol and next state. This table describes the transition of one state to another given an input symbol.
- A start state, which is a state in Q.
- A set of final states, F. F is a subset of Q.

In this thesis, only deterministic finite state automata (DFA) will be used. "Deterministic" denotes the fact that in every state, for every input symbol in Σ, there is one and only one state transition. "Finite" denotes the fact that the size of Q is finite.

State Machine Diagram Conventions

The following diagram shows the format of the state machine diagram used throughout this thesis.

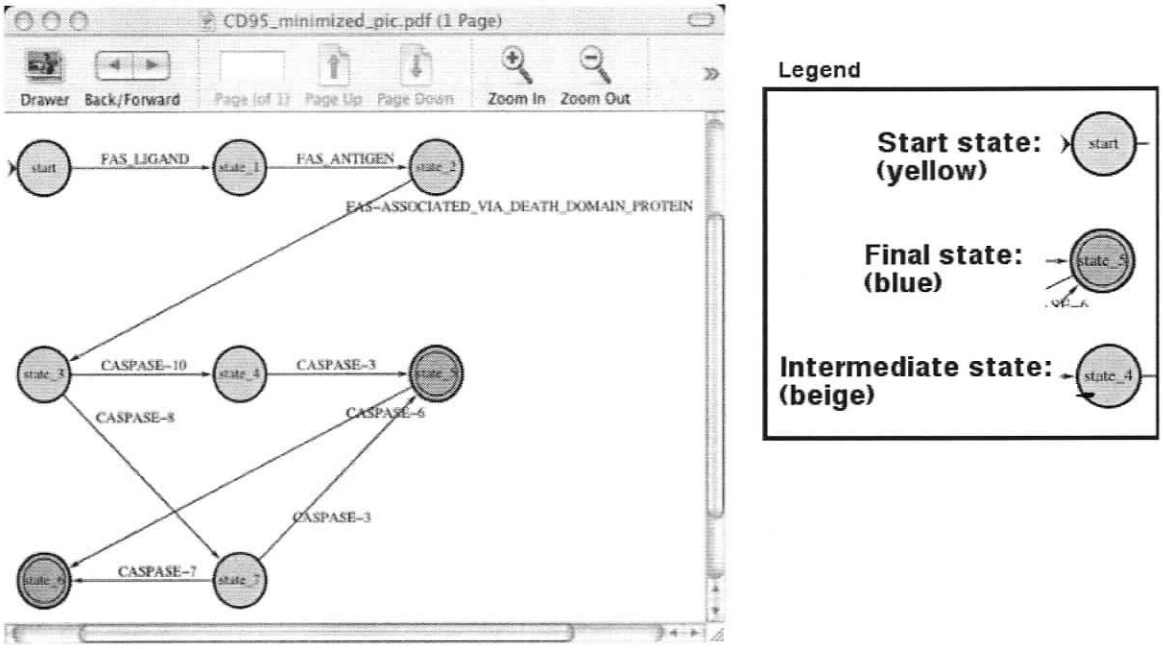


Figure 0-2: State machine diagram convention.

Dead States and DFA Missing Transitions [23]

A detail discussion of dead states and DFA missing transitions can be found on [23]. This is particularly important for this thesis as the pathway abstraction process does not define transitions for all input symbols for every state. When pathways are represented as DFA in this thesis, most of their states will contain missing transitions for most of the input symbols. In this case, input symbols with undefined transitions will transit to the dead state. The following figures show the presence of the dead state in pathway DFA's.

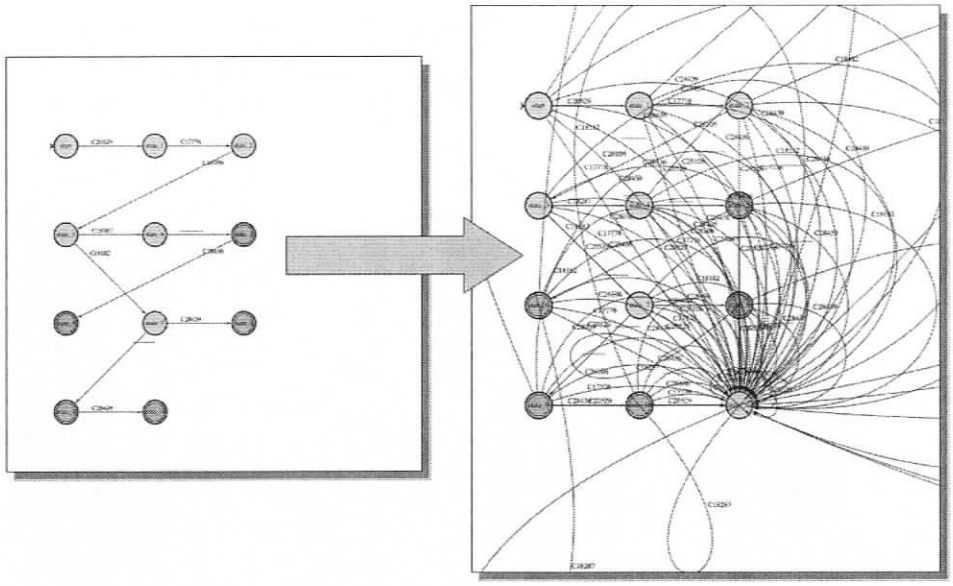


Figure 0-3: State machine with dead state (right) and without dead state (left).

For presentation purposes, all DFA diagrams in this thesis will not show transitions to the dead state. However, they are taken into account in the DFA analysis. In particular, they are needed for the DFA minimization algorithm. The only exception is found in section 4.1 "From Pathways to DFA without Cycles", where cycles to the dead state are allowed – i.e. the dead state is effectively ignored.

Longest Common Subsequence Problem

Please refer to [5] for a detail discussion on the longest common subsequence problem. The following is a formal definition of "subsequence" from [5].

> "Given a sequence $X = \langle x_1, x_2, \ldots, x_m \rangle$, another sequence $Z = \langle z_1, z_2, \ldots, z_k \rangle$ is a subsequence of X if there exists a strictly increasing sequence $\langle i_1, i_2, \ldots, i_k \rangle$ of indices of X such that for all $j = 1,2, \ldots, k$, we have $x_{i_j} = z_j$."

The longest common subsequence problem is to find the longest subsequence between two sequences. This problem is of particular interest to biologists because of its

application to pattern-matching in biological sequences such as DNA sequences. Algorithms have been developed to calculate mathematically optimal longest subsequence or alignments. These algorithms allow biologists to do (at least) the following:

- Assess an alignment objectively
- Automate searches for alignments

To find the longest common subsequence is trivial in a biological sense as it does not take into account any biological information and thus this method often does not yield true biological alignments. Biologists have been modifying this problem to more accurately reflect the underlying biology. For example, one method assigns different weights to different character matches/mismatches – e.g. PAM matrix [4].

Dynamic Programming

To find the longest common subsequence by exhaustively looking at every possible solution would take exponential computation time. Dynamic programming is an efficient algorithm to find the longest subsequence using n^2 time and n^2 space. An improved version, developed by Hirschberg [22], requires linear space. Please refer to [5] for a detail discussion on dynamic programming and a pseudo code of the n^2 time/space algorithm.

Chapter 1

1. Introduction

1.1. Pathway Comparison

Comparative analysis on biological sequences is a well-established academic discipline [9]. One of the main driving forces of this study is to help scientists learn how to interpret biological sequences. For example, one would like to know the functionality of a protein by just knowing its amino acid sequence. We would like to explore the idea of comparing pathways, in a manner similar to amino acid/nucleic acid sequence comparison, as a mean to further understand these pathways. For example, if pathways are similar, would they be regulated in a similar manner? If we observe reoccurrences of pathways that are similar, can this be evidence that such patterns in pathways have certain advantages? If so, it would be possible to learn from these patterns and apply them, for example, in areas of engineering. Existing works on pathway comparison deals mainly with metabolic pathways and regulatory pathways [2]. This thesis explores another aspect of biochemical pathway, namely, its molecular machinery – how molecules interact with each other to achieve a purpose. First, a model is developed to represent some aspects (i.e. an abstraction) of the pathway. Then, a framework is developed to compare pathways represented by instances of this model. This framework deals with issues such as how similar two model instances are, and what are their common features.

1.2. Example Results of Pathway Comparison

The following are some pathway comparison examples.

1.2.1. Apoptosis Pathway between Mammal and Yeast

Studies have shown that certain apoptotic pathways are very similar among very different species. In a review article, Meier *et al.* [33] describes these similarities. The following figures are taken from this article.

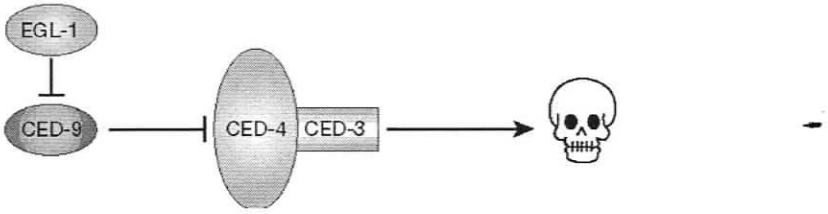


Figure 1-1: Caenorhabditis elegans (round worm) apoptotic pathway.

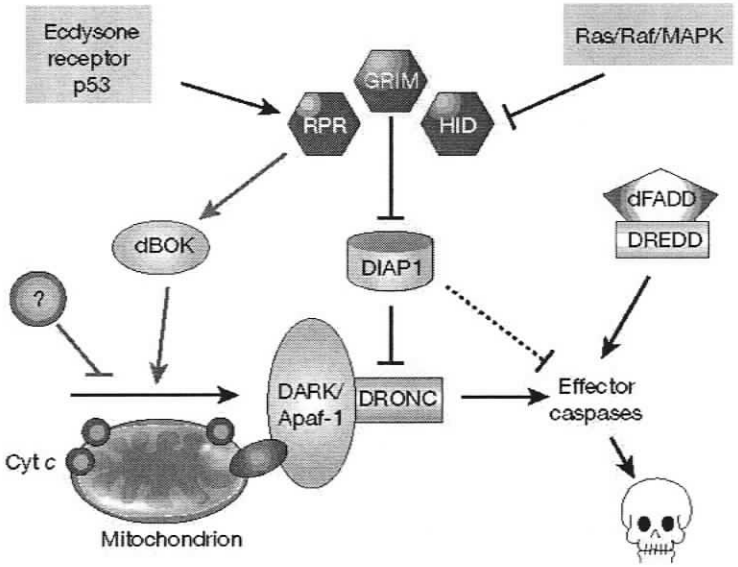


Figure 1-2: Fruit fly apoptotic pathway.

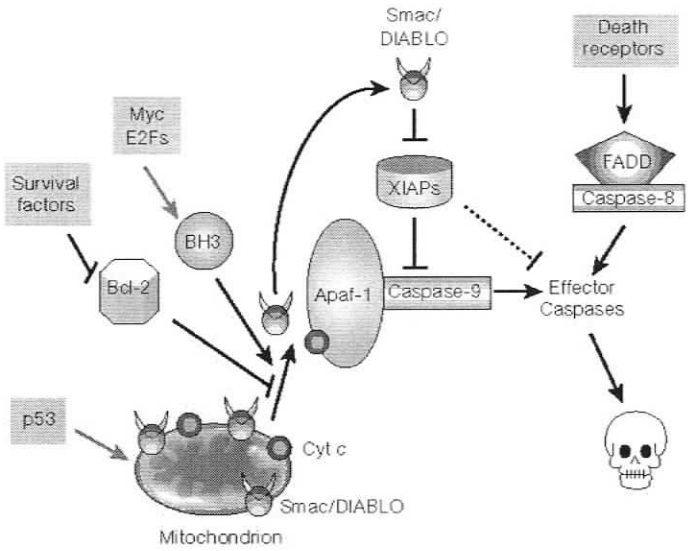


Figure 1-3: Mouse apoptotic pathway.

As can be seen from the above diagrams (Figure 1-1, Figure 1-2 and Figure 1-3), even though these apoptotic pathways differ very much in structure and complexity, some elements are very similar among them. The following table shows the homologous molecules among these apoptotic pathways.

	C. elegans	*Fruit fly*	*Mouse*
Homologous molecule 1	CED-4	DARK	Apaf-1
Homologous molecule 2	CED-3	DRONC	Caspase-9

Table 1: Homologous molecules in apoptotic pathway.

1.2.2. Glycolysis Pathway among Different Species

Dandekar *et al.* [7] did a detail comparison of the glycolytic pathways between species, including a number of bacteria and parasites (16 prokaryotes and yeast). They combined three different approaches in their pathways comparison: analysis and comparison of biochemical (pathway) data, algebraic analysis based on thermodynamic properties of the chemicals involved, and genome analysis. For example, genomes of different species

were compared to identify homologous / related glycolytic enzymes. Phylogenetic analyses were used. Hypothetical pathways where constructed using known thermodynamic properties of the different enzymes. These hypothetical pathways were used to explain how functions of "missing" enzymes (i.e. enzymes that appear in some species and not other) in some species could be accounted for. Their findings reveal a "surprising plasticity of the glycolytic pathway". [7] Although the different species share significant parts of the glycolytic pathway, some species were shown to have alternate pathways ("by-passes"). These alternate pathways are useful in the study of medicine and drug design. For example, a drug targeting a pathway of a parasite might be ineffective due to existence of alternative (by-pass) pathways.

1.2.3. Comparison between Biosynthesis of Different Amino Acids

Tohsato *et al.* [45] did a "comparative analysis of metabolic pathways based on similarity between enzymatic reactions to find pathway motifs which have reaction similarity in the pathways." In particular, they analyzed the different amino acid biosynthesis pathways. They represent amino acid biosynthesis pathways as strings of enzymes (i.e. the enzymes that are responsible for each chemical reaction in the pathway). They then align these strings of enzymes using dynamic programming. Their similarity matrix is calculated based on the Enzyme Commission [57] enzyme hierarchy. Their pathway alignment results "extract three groups of biosynthesis pathways of chemically-similar amino acids". The following table is taken out of their paper [45], which shows the three groups (clusters) of amino acid biosynthesis pathways found by pathway alignment.

Table 1: Common patterns and alignment scores obtained by the alignments between amino acid biosynthesis pathways.

His–Lys	15.0	[4.2.1] "–" "–" [2.6.1] [3]
His–Arg	12.8	[2.6.1] [3]
His–Pro	6.8	[2] [*] [1]
Lys–Arg	31.4	[2.7.2] [1.2.1] "–" "–" "–" [2.6.1] [3.5.1] "–" [*] [4]
Lys–Pro	19.7	[2.7.2] [1.2.1] "–" [1]
Arg–Pro	18.9	[2.7.2] [1.2.1]
Ile–Leu	25.6	[4] [4] [1.1.1] "–" [2.6.1.42]
Cys–Met	15.9	[2.3.1] [4.2.99]

Figure 1-4: Amino acid biosynthesis pathways group by pathway alignment. [45]

Please note the following in the table in Figure 1-4. The first column indicates which pathways are being aligned. For example, "His-Lys" represents the alignment between the biosynthesis pathway of histidine and lysine. The second column represents the alignment score. The third column represents the pathway alignment where the numbers (e.g. 4.2.1) represent classes of enzymes in the Enzyme Commission hierarchy and '-' represents a gap. Their experiments show that biologically relevant results (i.e. being able to cluster amino acid biosynthesis pathways into groups of biosynthesis pathways of chemically-similar amino acids) can be revealed by pathway alignment.

1.3. Contribution of this Thesis

This thesis introduces a quantitative framework for the general representation and comparison of biochemical pathways. First, this thesis introduces a methodology to abstract a biochemical pathway into a formal mathematical model. Then, some existing sequence comparison methods for comparing pathways are expressed by the proposed pathway model. This thesis incorporates the NCI Thesaurus into the pathway comparison process and show that a protein ontology, such as the NCI Thesaurus, can be useful in pathway comparison. Next, an extension to the proposed pathway model is

considered and methods to compare pathways expressed in this extended model are developed. More specifically, the contribution of this thesis is as follows:

1. Developed a model to represent certain aspect of biochemical pathways using deterministic finite automata without branches.
2. Developed a model to represent certain aspect of biochemical pathways using deterministic finite automata without cycles.
3. Showed that it is possible and useful to use protein ontology such as the NCI Thesaurus in pathway comparison.
4. Developed software tools to compare pathways in the following manners:
    - Pair-wise linear pathway alignment (linear pathways are deterministic finite automata with no branching structures).
    - Cluster linear pathways.
    - Multiple linear pathways alignment.
    - Pair-wise alignment for pathways expressed as deterministic finite automata with no cycles. This involved developing techniques for generating a "consensus deterministic automaton".

Chapter 2

2. Related Work

This chapter describes some related work.

2.1. Conserved Pathways within Bacteria and Yeast

Kelley *et al.* [27] [28] implemented "a strategy for aligning two protein-protein interaction networks that combines interaction topology and protein sequence similarity to identify conserved interaction pathways and complexes." They have developed a tool, PathBLAST, to perform protein-protein interaction networks alignment.

Their analyses involve the following. First, they extract linear paths from a protein-protein interaction network. This would give them a set of strings of protein names. In particular, consecutive proteins in these strings are shown, using biochemical methods, to interact with each other. They then aligned these strings of protein names using dynamic programming. The similarity matrix (similarity scores) is calculated based on the BLAST scores between the individual proteins – i.e. "blasting" one protein sequence with another protein sequence. The following figure is taken from their paper [27], which shows their alignment procedure.

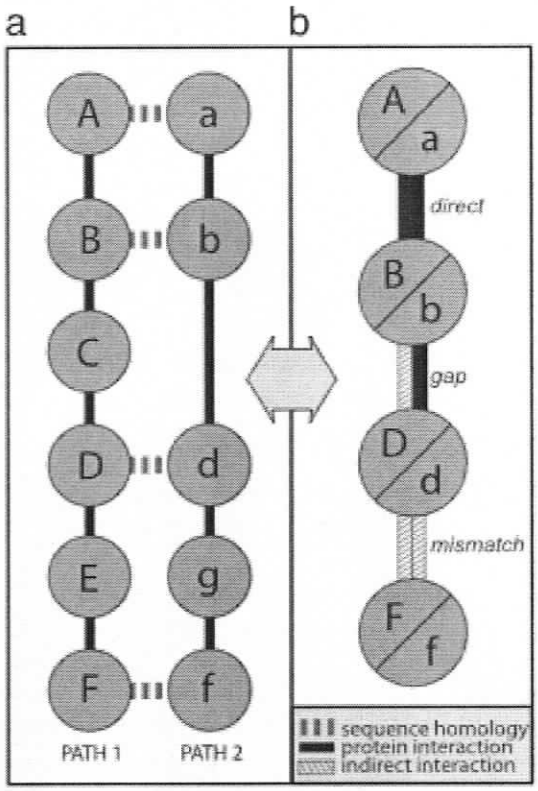


Fig. 1. Example pathway alignment and merged representation. (a) Vertical solid lines indicate direct protein–protein interactions within a single pathway, and horizontal dotted lines link proteins with significant sequence similarity (BLAST E value $\leq E_{cutoff}$). An interaction in one pathway may skip over a protein in the other (protein C), introducing a "gap." Proteins at a particular position that are dissimilar in sequence (E value $> E_{cutoff}$, proteins E and g) introduce a "mismatch." The same protein pair may not occur more than once per pathway, and neither gaps nor mismatches may occur consecutively. (*b*) Pathways are combined as a global alignment graph in which each node represents a homologous protein pair and links represent protein interaction relationships of three types: direct interaction, gap (one interaction is indirect), and mismatch (both interactions are indirect).

Figure 2-1: Pathway alignment procedure used by PathBLAST [27]

They did the following experiments. They aligned a yeast (Saccharomyces cerevisiae) protein-protein interaction network against itself and they aligned a yeast protein-protein interaction network against a bacteria (Helicobacter pylori) protein-protein interaction network. Their pathway alignment experiments have led to some very interesting biologically relevant insights such as the following:

1. Pathways from a well-studied network (yeast) have "shed light on their aligned counterparts from a less well-characterized one" (bacteria – H. pylori). For example, some unknown proteins in H. pylori were aligned to well-studied proteins in yeast, and hence the function of these unknown bacterial proteins can be deduced.
2. Pathway alignments can link "two or more pathways or cellular processes not previously known to associate". They were able to show previously unknown cross talks between pathways using pathway alignments.
3. "Single pathways in bacteria frequently correspond to multiple pathways in yeast …"
4. Proteins "within high-scoring pathway alignments did not necessarily pair with their best sequence matches in the other pathway." This shows that pathway alignment can provide additional information that complements sequence alignment information in determining functions of proteins.

2.2. Metabolic Pathway Analysis using Enzyme Hierarchy

Tohsato *et al.* [45] did a "comparative analysis of metabolic pathways based on similarity between enzymatic reactions to find pathway motifs which have reaction similarity in the pathways." In particular, they analyzed the different amino acid biosynthesis pathways. They represent amino acid biosynthesis pathways as strings of enzymes (i.e. the enzymes that are responsible for each chemical reaction in the pathway). They then align these strings of enzymes using dynamic programming. Their similarity matrix is calculated based on the Enzyme Commission [57] enzyme hierarchy. Please refer to section 3.3.2.3 of this thesis for a detail explanation on how the similarity matrix (similarity score) is calculated. Their results are briefly described in section 1.2.3 of this thesis.

Chapter 3

3. Part I: Abstraction to Linear Pathways

This chapter describes the representation of biochemical pathways as a sequence of protein names.

3.1. Motivation for Abstraction

Why abstract? Ideally, one would like to construct a model to capture all aspects of biochemical pathways – all the properties that would allow one to simulate a pathway *in silico*. Attempts to do so include the use of process algebra [38]. However, there are some difficulties:

- The kinetics of very few pathways is known to the detail of every interaction between different molecules. Therefore, models that capture all aspects of biochemical pathways can be constructed for only a small number of pathways.
- Even if the kinetics of many pathways were known, analysis of such model would be computationally very expensive.
- Even if computation power is not a factor, no mathematical framework exists currently that allows a formal comparison between pathways expressed as, for example, process algebra. Concurrency theory research deals with the notion of equivalence (e.g. trace equivalence, bisimulation). Such analyses allow one to consider if the two systems are equivalent, but not how different they are when they are not equivalent.

Therefore, when doing comparative studies on pathways, one is actually comparing certain aspects of biochemical pathways. This brings up the concern of how well the model reflects the reality. Can pathways be represented as an abstraction and at the same time, the reasoning done on their properties reflect the actual biological process? The simple answer is: it is not known for sure but perhaps. This issue is common with all mathematical modeling exercises. It is needed to first prove that the model reflects certain aspects of the real-world system and infer, based on the fact that the model reflects observed behaviors of the system, that the model can then be used to predict behaviors of the system that have not yet been observed.

From a practical point of view, scientists have been collecting data on pathways for over a century and the amount of such data is enormous. Without a model that can incorporate these data, scientists would have no way of capturing these data into a framework that would allow knowledge to be extracted. In this case, this "knowledge" can be a further understanding of the behavior and properties of the pathways, an ability to draw conclusion about unobserved behavior of the pathways, or a classification of pathways, which could give insights as to how a particular pathway fits into the "whole picture" of the biological system.

A very successful abstraction model is found in Chemistry. Lewis structure is an abstraction model used to represent atoms and chemical interactions between atoms. It provides a way to reason and explain various chemical reactions. The model focuses on the valance electrons of the atoms. It did not account for the orbital of the elections[3], or any quantum mechanics of the atoms. As a result, different assumption rules and

[3] Orbital is yet another model to describe atoms.

exceptions are needed to explain a great number of chemical reactions. The success of the model lies on its simplicity both for understanding certain aspects about atoms and for reasoning about simple chemical reactions. Zumdahl [48] provides a detailed explanation of Lewis structure. It is desired to be able to develop a model for describing and reasoning about biochemical pathways that is simple enough to allow the representation of large number of biochemical pathways but does not have to worry about missing/unknown data, yet at the same time is accurate enough to allow constructive reasoning about pathways. As mentioned before, this thesis focuses on the molecular machinery aspect of biochemical pathways – i.e. how molecules physically interact with each other.

3.2. From Pathways to DFA without Branching to Strings (Linear Pathways)

Consider a chemical reaction occurring in a black box (e.g. a test tube, a compartment in a cell). When the chemicals in the black box such as a test tube, are in equilibrium, one can consider that this is a state. This is because without the introduction of any "action" such as introduction of chemicals, introduction of heat, or degradation of chemicals, the concentration of all chemicals in this black box will remain the same indefinitely. When a chemical is introduced to the black box and thus initiated a chemical reaction, the chemicals in the black box will try to reach for another equilibrium – another state. Hence, theoretically, a state machine can be used to describe the behavior of a chemical system such as a cell. It is very computational expensive to define a state that takes into account all chemicals in the cell. However, a state can be defined that takes into account only a certain number of chemicals.

The effects of the introduction of a chemical species are dependant on the amount of chemicals introduced. This is the thermodynamic feature of the chemical system. One could also consider the kinetic aspect of the chemical system, e.g. how much time does it take the chemical system to reach another equilibrium. The thermodynamics and kinetics are not available for many reactions that occur in the cell suggesting that they are not known[4]. Hence, it would be ideal to develop a model (state machine) that would not need to rely on all thermodynamics and kinetics data. Yet, it is necessary to take into account, at the minimum, the stoichiometry of the chemical reactions. This thesis proposes an abstraction model that consists of the following.

- A state is defined as a set of ranges of concentrations for a set of chemical species in the cell. For example, assume that the model deals with chemical species A, B, and C. A state, *S*, can be defined as when the concentration of A is between 0.11 μmol/L and 0.12 μmol/L, the concentration of B is between 0.01 μmol/L and 0.02 μmol/L, and the concentration of C is between 0.01 μmol/L and 0.05 μmol/L. It is not required that the chemical species in a state be at equilibrium. During pathway analysis, the actual concentrations of the chemical species in all the states are not considered – they are only assumed to exist with certain finite values. In fact, the actual numbers are not known.

[4] The author of this thesis knows this because if these are known for many chemical reactions in the cell, there would be public databases available (similar to KEGG) that describes the thermodynamics and kinetics constants of the different chemical reactions in the cell. As for now, in KEGG, the thermodynamic nature of the pathways is represented using diagrams and no kinetic information is available. In fact, developing a standard model/syntax to allow the exchange of pathway information is still under active research/discussion [56].

- An input action is a protein name. The analysis in this thesis only deals with situation where input action is a protein. However, the methodology can be extended to include non-protein such as RNA's and inorganic molecules in a straightforward manner. When an input action occurs, it is assumed that the protein introduced is of sufficient concentration to induce changes in the concentration of different chemical species via chemical reactions to "move" the cell to the next state. Please note that the "next state" could be the same as the "current state". The actual protein concentration associated with an input action is not considered during pathway analysis – this number is only assumed to exist. To simplify the model further, inputs of the same protein but with different concentrations are considered as the same input.

The state machine representation of biochemical pathway is defined. However, it is still too complicated for the first attempt at pathway comparison. Thus, for the first attempt, only strings with finite length represented by the state machine are considered – i.e. strings of protein names, one string at a time. This amounts to ignoring all the branching structure of a state machine and only analyzes a single sequence of input actions.

The following is an example of how to construct an abstraction of a biochemical pathway using a string of protein names. Consider the apoptosis pathway as shown in the following diagram taken from KEGG [26]. Please refer to Appendix B. "KEGG Diagram Legend" for the legend of this figure.

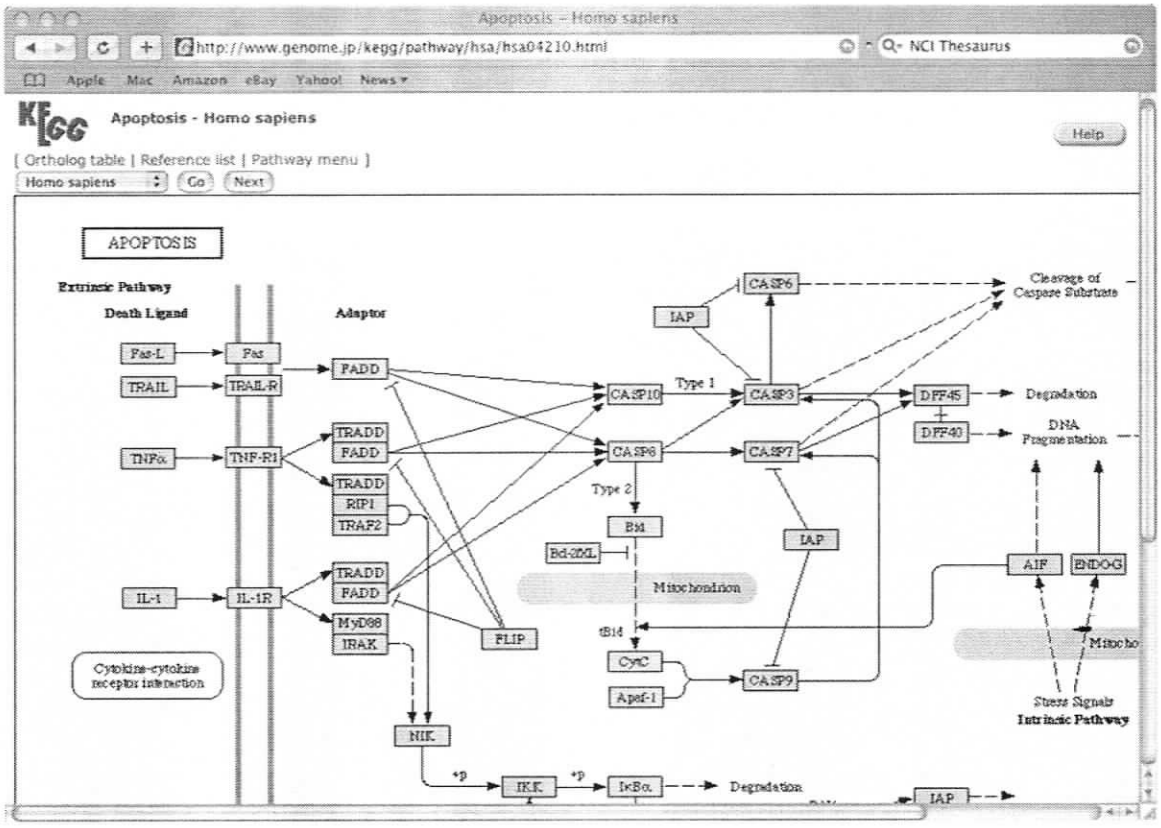


Figure 3-1: Apoptosis diagram from KEGG.

A portion of the above pathway, the signaling cascade initiated by "Fas-L", can be represented using a state machine as shown in the following figure. For the state machine diagram conventions used throughout this thesis, please refer to Figure 0-2:State machine diagram convention.

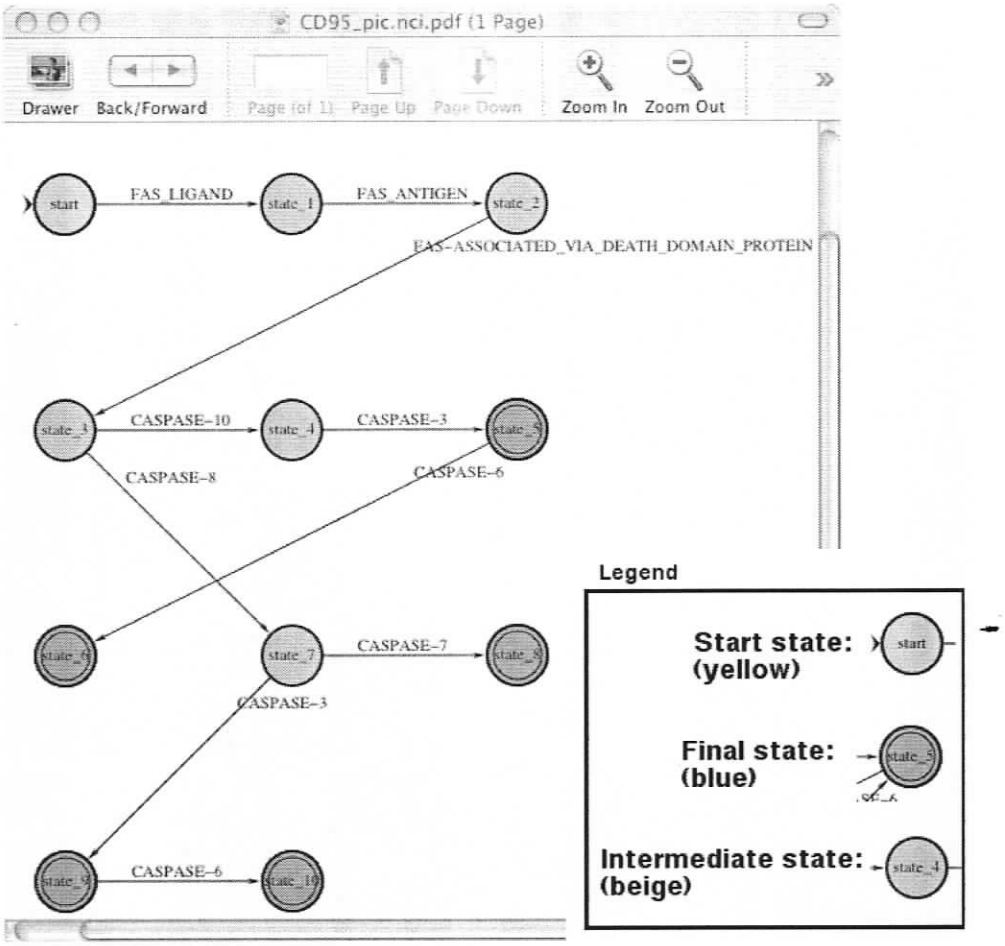


Figure 3-2: State machine representing apoptosis initiated by "Fas-ligand".

If only one particular path is considered from the "start" state to a final state, a state machine may be deduced similar to the following.

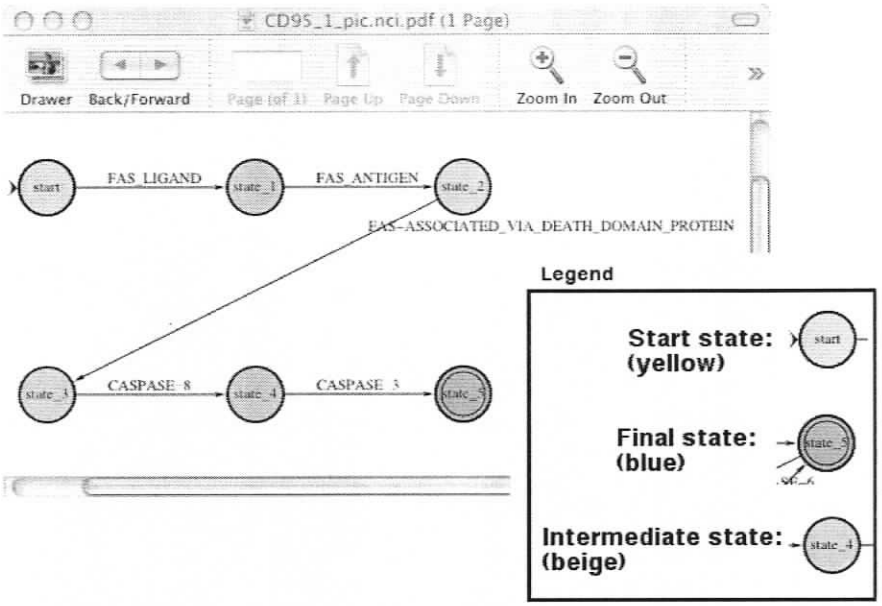


Figure 3-3: State machine representing one path from the "start" to a final state.

This is the biochemical pathway abstraction/representation used for the first attempt in pathway comparison analysis. In this thesis, the above state machine is referred to "state machine or deterministic finite automata (DFA) without branching" or "linear pathway". Because this state machine represents only one string, it is more convenient to refer to this biochemical pathway abstraction as a string of protein names:

```
FAS_LIGAND FAS_ANTIGEN FAS-ASSOCIATED_VIA_DEATH_DOMAIN_PROTEIN CASPASE-
8 CASPASE-3
```

3.3. Similarity Matrix

Since it is possible to represent pathways as strings, pathway comparison can be performed using string comparison – i.e. the longest common subsequence problem. The proposed comparison method would do very little if one only considers subsequences of exactly matched elements. Because proteins have very specific functions, most of them only appear in one or a few related pathways. Reoccurring patterns (motifs) in a pathway

would most likely take the form of sequences of classes of proteins. The scoring model would therefore need to deal with exact match, partial match and mismatch, and gap (insertion/deletion). Gap cost (penalty) is calculated using the most basic model with cost-free end gaps: *cost of gap = constant * number of gaps*. The constant is arbitrarily chosen. Please refer to section 3.3.2.4.3 "An Explanation of Gap Cost". The cost of a match, partial match or mismatch is determined by the similarity between two elements (i.e. proteins). A tool that functions as a similarity matrix is needed.

3.3.1. Hierarchy of Input Symbols

For the analysis, this thesis restricts the scope to deal only with proteins. Determining the similarity between proteins can be done in one of the following ways:

- Sequence similarity – this is the most objective criterion for comparing proteins. However, proteins with similar biological functions can have very low sequence similarity.
- Structural similarity – this can reflect more accurately how proteins are similar/different functionally because proteins' biochemical function (e.g. enzymatic activity) is largely determined by the proteins' three-dimensional structure. However, structural data, as determined by physical experiment such as x-ray crystallography, are available for only a small number of proteins. Structural information for most proteins comes from matching the protein's sequence with known consensus sequences for structural domains. These computational methods may not reveal all relevant domains accurately. Even if all structural domains can be predicted or determined, one would still need to define a classification of structural domains such that one could determine how

similar two domains are. This is because structural domains, especially those that are involved in biochemical reactions (e.g. enzyme's active site), can be very specific. Therefore, proteins that perform similar function may have no common structural domains.

- Functional similarity – here, function pertains to biochemically-determined function. Large amounts of information concerning proteins' function is available in public databases. For example, the National Cancer Institute (NCI) has developed an ontology-like classification system (NCI Thesaurus) for proteins that are involved in cancer research [16]. This classification system is developed manually via consultation with existing biology/biochemistry literature. This classification system therefore represents the "current" understanding of how proteins differ from each other based on their biochemical functions. One can then calculate the similarity between two proteins based on their location in the ontology tree using the method in [45]. This method is advantageous because it takes into account "all" available knowledge in the scientific community. At the same time, this method can be very "biased", because it is only as good as what the current scientific community suggests, which may be completely wrong! Also, because not all proteins in the universe are classified in this ontology, it would show a more stringent comparison between proteins in classes that are more well-known than proteins in classes that are lesser known. For example, consider protein A and B in a class of proteins with a large number of proteins known to be in this class and protein C and D in a class with a smaller number of known proteins. The ontology would give the impression that C and D are more

similar than A and B because C and D are in a smaller sub-tree than A and B. However, this might not be true at all. This is an inherent problem of the method in [45] because it assumes that proteins in smaller sub-tree are more similar than proteins in larger sub-tree.

Despite its limitations, we believe that using a protein classification system, such as the NCI Thesaurus, is currently the only method to allow efficient and meaningful functional comparison of proteins in general (i.e. not just comparison between "related" proteins). To illustrate this point, five proteins were compared using the three methods mentioned above.

Protein Description	**Genbank Accession Number (mRNA transcript)**	**Enzymatic Function**
Adenosine Deaminase, Homo Sapiens	NM_000022	Hydrolase
Ap3A-Hydrolase (FHIT: fragile histidine triad gene), Homo Sapiens	NM_002012	Hydrolase
Cytochrome P450 Reductase, Homo Sapiens	NM_002012	Oxidoreductase
Aldehyde Dehydrogenase 1 family, member A2 (ALDH1A2), Homo Sapiens	NM_003888	Oxidoreductase
Topoisomerase DNA I TOP1, Homo Sapiens	NM_003286	Isomerase

Table 3-1: Protein Comparison.

The amino acid sequences of the above proteins were compared using a guide tree generated by clustalw [44]. A guide tree is a hierarchical clustering (represented by a binary tree) of the protein sequences based on a sequence similarity-based distance scoring system. Clustalw constructs this tree by first generating a distance matrix of the input protein sequences. The distance scored is calculated using the model of Kimura [30] based on sequence similarity scores determined by dynamic programming. A neighbor-joining clustering algorithm [41] is then used to generate a binary tree. The

following diagram (dendrogram) shows this guide tree. This dendrogram was generated by NJplot [35].

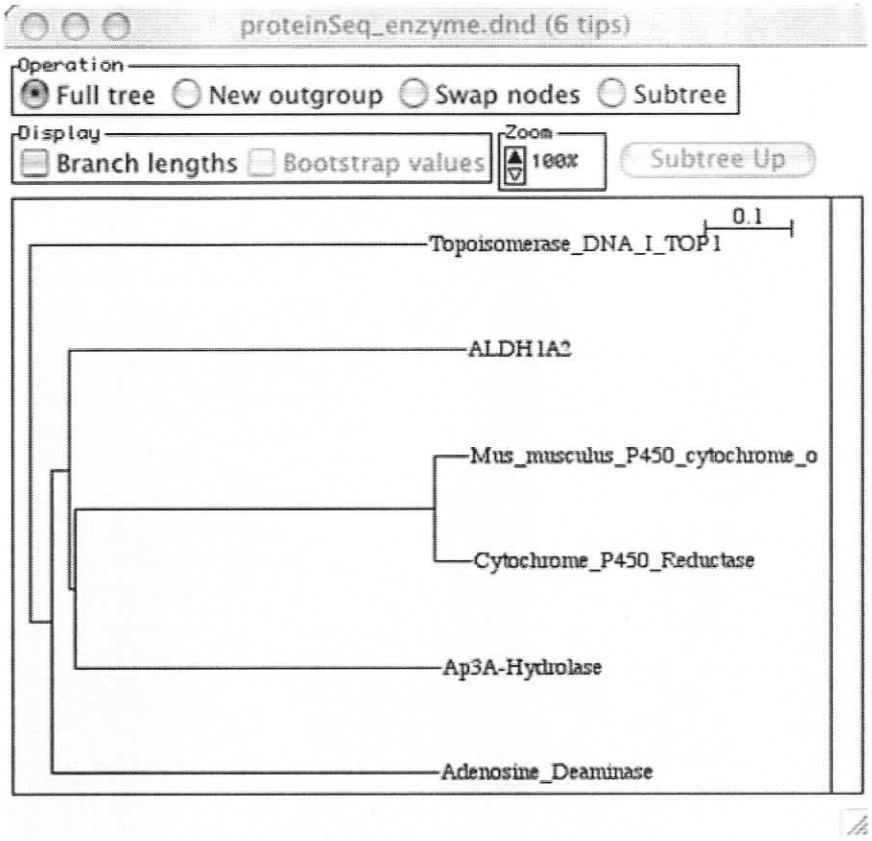


Figure 3-4: Clustalw guide tree drawn using NJplot.

It can be observed that Clustalw's guide tree did not cluster any of the input proteins into their functional group (hydrolase, Oxidoreductase and Isomerase). It seems that not much information can be extracted from the guide tree because the branch lengths for all the proteins are fairly much the same – i.e. to clustalw they all look similar. To give one an idea as to how similar these proteins are, based on sequence alignments, the mouse (Mus Musculus) version of the cytochrome P450 reductase, which has 91% identical sequence with the human counterpart, was included in the alignment. Immediately, it can be seen that this protein aligns well with human's cytochome p450 reductase.

If the domains of the proteins were compared, maybe this would reflect more accurately how the proteins' functional capabilities differ. One of the largest protein domain databases available, Pfam [3], was used for searching for the known domains of the input proteins shown in Table 3-2. The following table shows the Pfam domain matches with high statistical significance (i.e. trusted).

Protein Name (column) / Pfam Domain Name (row)	Adenosine Deaminase	Ap3A-Hydrolase	Cytochrome (Human)	Cytochrome (Mouse)	ALDH1A2	Topoisomerase DNA I TOP1
A deaminase	**YES**	*NO*	*NO*	*NO*	*NO*	*NO*
HIT	*NO*	**YES**	*NO*	*NO*	*NO*	*NO*
Flavodoxin 1	*NO*	*NO*	**YES**	**YES**	*NO*	*NO*
FAD binding 1	*NO*	*NO*	**YES**	**YES**	*NO*	*NO*
NAD binding 1	*NO*	*NO*	**YES**	**YES**	*NO*	*NO*
Aldedh	*NO*	*NO*	*NO*	*NO*	**YES**	*NO*
Topoisom I N	*NO*	*NO*	*NO*	*NO*	*NO*	**YES**
Topoisom I	*NO*	*NO*	*NO*	*NO*	*NO*	**YES**

Table 3-2: Trusted Pfam domains.

It turns out that none of the proteins has any domains in common, except the two cytochromes. Even for lower statistically significant Pfam domain matches, none of the proteins (except the cytochromes) has any domains in common.

Lastly, the NCI Thesaurus gives the protein classification as shown in Table 3-1. Therefore, for the pathway comparison purpose in this thesis, it is most efficient to use the NCI Thesaurus to determine protein function similarities.

3.3.2. NCI Thesaurus

The subsection briefly describes the NCI Thesaurus and how it is used in the biochemical pathway analysis done in this thesis.

3.3.2.1. Overview of Ontology

Please refer to [18], [34] and [65] for a more detail discussion on ontology. The following are two definitions of "ontology":

> "*A specification of a representational vocabulary for a shared domain of discourse — definitions of classes, relations, functions, and other objects...*" [18]

> "*An ontology is a specification of a conceptualization*[5]." [65]

Ontology describes each term in a vocabulary with a specific "meaning" or definition. This can involve identifying the attributes a term can have and setting constraints on these attributes – e.g. a dog must have exactly four legs, where "dog" is a term, "have legs" is an attribute, and "have exactly four" is a constraint. Ontology also specifies the relationship between all the terms in that vocabulary – e.g. a dog is a kind of animal, where "dog" and "animal" are terms and "a kind of" is a relationship.

One of the main usages of ontology is to allow agents (e.g. a computer program) to communicate with other agents using a controlled vocabulary (i.e. an ontology). This would allow the agents to reason about exchanged information based on a common context (i.e. speak the same language). Some recent research areas that use ontology include the semantic web [64] and the usage of classifiers in medical research [37][6]. Another usage of ontology is to define taxonomic hierarchies of classes. One can treat an ontology as a classification system of terms. In this case, one would be mainly dealing

---

[5] "Conceptualization" is "an abstract, simplified view of the world that are represented for some purpose."[65]

[6] Rector's presentation is available at http://protege.stanford.edu/conference/2004/slides/6.3_rector_Why_classify_Protege_workshop_2004.pdf. His research involves building ontologies to describe medical conditions (e.g. diabetes). These ontologies can then be used to classify patients. Based on the group that the patients are classified into, one can use these ontologies to infer other medical conditions these patients may have or have a high risk in having.

with the "is-a" relationship among the terms in the ontology – similar to the subclass – class – super-class relationship in object design in object-oriented programming.

3.3.2.2. Description of NCI Thesaurus

The NCI Thesaurus is a public domain description-logic-based terminology produced by the National Cancer institute [16]. It is a classification of the technical terminologies used by the NCI. One of the reasons why this thesaurus (or more appropriate, its predecessor, the NCI Metathesaurus) was constructed is to allow the NCI staff to annotate, document and/or communicate their research activities using a controlled vocabulary. Such documentation would allow one to monitor, manage and analyze the projects within NCI efficiently. Even though the NCI Thesaurus contains many features of an ontology, it is not a true ontology. It may be described as "a nomenclature with ontologic features." [16] One part of the NCI Thesaurus that is particularly useful for the pathway comparison analysis done in this thesis is a classification of proteins involved in cancer research. Proteins are classified under four different classification systems:

- By location – e.g. cytoplasmic protein (proteins that are in the cytoplasm), extracellular protein (protein that are outside the cell).
- By origin – e.g. fusion protein ("A protein created by joining two genes together. Fusion proteins may occur naturally or can be created in the laboratory for research."[7]) and non-human protein (proteins that are not from human).
- By structure – e.g. glycoprotein ("A protein that has sugar molecules attached to it."[8]) and metalloprotein ("Proteins that have one or more tightly bound metal ions forming part of their structure."[9])

---

[7] This is one of the definitions of "fusion protein" in the NCI Thesaurus [52].

- By function – e.g. enzyme ("A protein that speeds up chemical reactions in the body."[10]) and growth factor ("Substances made by the body that function to regulate cell division and cell survival. Some growth factors are also produced in the laboratory and used in biological therapy.[11]")

Since the pathway comparison analysis done in this thesis is concerned about the biological functions of proteins, it uses the classification system based on the protein's function.

3.3.2.3. Similarity Score Calculation

How is a protein classification system used in the pathway comparison analysis? It is used exactly the same way as Tohsato et al. uses the Enzyme Commission (EC) number [57] classification of enzymes in [45] and [46]. EC number is a four-integer identifier that is assigned to different classes of enzyme based on their biochemical functions[12]. It was originally assigned by the "International Commission on Enzymes", which was established in 1956. Currently, the EC number classification system is maintained by the "Nomenclature Committee of the International Union of Biochemistry and Molecular Biology" (NC-IUBMB). Please refer to [58] for further information on the history of EC numbers. The format of EC number is w.x.y.z (similar to IPv4 address format with the difference of w, x, y, and z having a range of 0 or greater instead of 0 to 255). For

---

[8] This is one of the definitions of "glycoprotein" in the NCI Thesaurus [52].
[9] This is one of the definitions of "metalloprotein" in the NCI Thesaurus [52].
[10] This is one of the definitions of "enzyme" in the NCI Thesaurus [52].
[11] This is one of the definitions of "growth factor" in the NCI Thesaurus [52].
[12] The original goal of the EC number classification system is to "*consider the classification and nomenclature of enzymes and coenzymes, their units of activity and standard methods of assay, together with the symbols used in the description of enzyme kinetics.*" [58]

example, glutamate synthase (NADPH) is assigned an EC number of 1.4.1.13 and Trypsin is assigned an EC number of 3.4.21.4. In determining the function of the enzyme, the left-most integer of the EC number is most significant – i.e. enzymes that differ by the left-most integer of their EC number have the most functional differences, while the ones that differ by the right-most integer of their EC number have the least differences in their biochemical function. It is then intuitive to represent the EC number enzyme classification system as a tree where the leaf nodes are the enzyme classes with a 4-integer EC number, and the internal nodes are the "superclasses" of enzymes – e.g. 3.4.11.x is a class of "all" aminopeptidases. Tohsato et al. developed a scoring system that uses the EC classification system to determine how different two enzymes are. The statistic they used is given by the following equation.

$$I(h) = -\log\frac{C(h)}{C([*])} - \log p(h)$$

Equation 1: Similarity scoring system for enzymes.

where

- h = "lowest" common superclass (parent) of two enzymes
- $I(h)$ = similarity score
- $C(h)$ = number of enzymes in class h
- $C[*]$ = number of enzymes classified in the EC number classification system
- $p(h) = o(h)/N$, where
    - o(h) = is the number of enzymes that belong to class h
    - N = the total number of enzymes in the two pathways being aligned

The above equation functions as the "similarity matrix".

In this thesis, we would like to consider proteins in general. If we consider the NCI Thesaurus (the protein classification system) as a tree, we can use the above equation (Equation 1) with the following changes.

$$I(h) = -\log \frac{C(h)}{C([*])} - \log p(h)$$

Equation 2: Similarity scoring system for proteins.

- h = "lowest" common superclass (parent) of two proteins
- $I(h)$ = similarity score
- $C(h)$ = number of proteins in class h
- $C[*]$ = number of proteins in the NCI Thesaurus
- $p(h) = o(h)/N$, where
    - o(h) = is the number of proteins that belong to class h
    - N = the total number of proteins in the two pathways being aligned

3.3.2.4. Rationale for the Similarity Score Calculation

The intuition behind the similarity scoring system (Equation 2) can be understood from two perspectives – from a biological function/characterization perspective and from an information content perspective.

3.3.2.4.1. Biological Function/Characterization Perspective

In terms of biological function and characterization, there are two terms in Equation 2. The first term represents the fraction of proteins that belong to the lowest common protein class containing the two proteins being compared. This shows the similarity between the two proteins based solely on the ontology. The "log ()" is used so that the

similarity scores can be additive. However, as suggested in [46], similarity scores based solely on the ontology could be misleading. This is because the ontology might not be complete. Consider a very specific protein class that is well studied. There might be a large number of proteins known to belong to this class. The first term in Equation 2 would give a misleading similarity score between proteins in this class – the score would suggest that these proteins are not "that similar" because proteins in this class represents a relatively large portion of proteins in the ontology. For example, as mentioned in [46], there are 254 classes on enzymes that has EC number starting with [1.1.1] but there is only one enzyme class that has EC number starting with [5.3.4].

The second term in Equation 2 is an attempt to compensate the problem mentioned above. This term (referred to as "occurrence probability" in [46]) is the number of proteins in the pathways being compared that belongs to the lowest common protein class (between the proteins being compared) over the total number of proteins in the pathways being compared. This term shows how general or specific the lowest common protein class is based on the "protein population" in the pathways being aligned. Consider a very specific protein class that has many known members. It is unlikely that the many proteins in the pathways being aligned belong to this class. Thus, the second term in Equation 2 has a significant similarity score even though the first term might have a less significant score. This is the compensation effect. Therefore, the second term of Equation 2 assumes that protein classes are evenly distributed in all pathways.

3.3.2.4.2. Information Content Perspective

From the information content point of view, one could think of the "best" pathway alignment as an alignment with the highest information content – an alignment that

would show the most about the common features between the two pathways. In addition to the information content of the final alignment string, each protein "symbol" is given an intrinsic information content value. These information content values reflect the similarity between proteins according to the NCI Thesaurus – i.e. how far apart are the two proteins located in the NCI Thesaurus ontology tree. The farther away the two proteins are, the lower the information content. The intuition behind this scoring scheme is as follows: when two proteins are far away within the ontology tree (i.e. their lowest common parent protein class is large), less information about the protein is known. For example, consider when a protein is referred to as an "enzyme" comparing to when it is referred to as a "caspase". In the latter case, one knows more about the protein than in the former case – in addition to knowing that the protein is an enzyme, one also knows that it is a hydrolase, in particular, a protease, and in particular, a cysteine proteinase. Thus, specifying a protein as a "caspase" amounts to having more information being encoded than specifying a protein as an "enzyme". The first term of Equation 2 is the intrinsic information content value of the proteins (more appropriately, protein classes) in the alignment string. The alignment algorithm tries to maximize the sum of the intrinsic information content of all proteins in the alignment string. Trying to maximize the information content of the alignment string in this respect would result in an alignment that maximizes the specificities of protein classes in the alignment string.

The second term in Equation 2 is the information content of the alignment string itself. The protein symbols in the two pathways represent the alphabets. The information content of the alignment string is calculated based on the entropy of the protein symbols in the alignment string. "Based on the entropy, information theory shows how to

calculate the probability of any string from the alphabet and predict its best compression, i.e., the minimum number of bits needed, on average, to represent the string." [43] The best compression estimation, the number of bits needed to represent this alignment string, is used as an optimization criterion in the alignment algorithm. This is because the compression estimate is an indication of the information content of the alignment string. To maximize the compression estimate is equivalent to trying to find an alignment string that contains the most information – i.e. shows the most about the common features between the two pathways. The following equation is used to calculate the compression estimate of the alignment string.

$$b(s) = -\left| \log_2 \left(\sum_{i=1}^{n} P_i \right) \right|$$

Equation 3: Compression estimation of an alignment string, *s*.

- $b(s)$ = the compression estimation (number of bits) of an alignment string, s.
- P_i = the occurrence probability of the protein symbol at the i^{th} position of s.

The occurrence probability of a protein symbol (protein class) is defined to be the percentage of proteins in the two pathways being aligned that belongs to that protein class. For example, consider the following two pathways:

FAS_L → FAS → FADD → Caspase-8 → Caspase-3

FAS_L → FAS → FADD → Caspase-8 → Caspase-7

The occurrence probability of the "enzyme" protein class would be: $P_{enzyme} = 4/10 = 0.4$ since there are four enzymes (Caspase-8, Caspase-8, Caspase-7, and Caspase-3) and ten proteins in the above two pathways. This criterion favors alignment strings with proteins matches (protein classes) that are relatively uncommon among the proteins in the

pathways being aligned (i.e. small $P_{protein\ class}$). This does make biological sense if one reasons as follows: A pathway is an ensemble of interoperating biochemical reactions and, because biochemical reactions are often very specific, biochemical reactions within a pathway should be diverse enough so that they could complement each other to carry out the overall function(s) of the pathway.

Gaps in the alignment string are assigned an arbitrary score. Please refer to section 3.3.2.4.3 "An Explanation of Gap Cost".

One must keep in mind that the "true" alignment might not be the alignment with the highest alignment score as with all alignment algorithms for biological sequences. The alignment criteria imposed by the similarity scoring system might be biologically valid for some cases but certainly not for all cases. For example, the "correct" alignment needs not to have the largest information content among all possible alignments.

3.3.2.4.3. An Explanation of Gap Cost

Gaps in alignment strings are assigned a penalty score = *constant* × *number of gaps*. The *constant* (gap penalty) is chosen so that the alignments generated by the algorithm are biologically meaningful. Taking too high of a gap penalty discourages the algorithm to take any gap and so a true gap would be replaced by a mismatch. Taking too low of a gap penalty encourages the algorithm to take too much gaps so a true mismatch would be replaced by a gap. The right balance would allow the algorithm to pick up gaps and mismatches that would make most biological sense. Thus, in one sense, gap cost is acting as a fudging factor to tune the alignment algorithm. In another sense, the gap cost that allows the alignment algorithm to generate the most meaningful alignments reflects the true biological gap cost.

To appreciate how gap cost influences alignments, consider three alignments of the following two linear pathways, each alignment using a different gap cost.

Pathway Description	Protein Names							
Pathway 1: Apoptosis cascade induced by FAS-ligand.	FAS Ligand	FAS Antigen	----------	----------	FAS-Associated Via Death Domain Protein (FADD)	Caspase-8	Caspase-3	----------
Pathway 1: Apoptosis cascade induced by Tumor Necrosis Factor-alpha (TNFα).	TNFα	TNFα Receptor	TNF Receptor-Associated Protein with a Death Domain (TRADD Protein)	Receptor-Interacting Serine-Threonine Kinase-1	TNF Receptor-Associated Factor-2 (TRAF2)	Serine-Threonine Protein Kinase-NIK	Conserved Helix-loop-helix Ubiquitous Kinase	I-Kappa-B-Alpha Protein
Aligned (Consensus) Pathway.	TNF Family Protein	TNF Receptor Family Protein	----------	----------	Adaptor Signaling Protein	Enzyme	Enzyme	----------

Table 3-3: Pair-wise pathway alignment – gap cost, -1×10^{-15}.

Pathway Description	Protein Names							
Pathway 1: Apoptosis cascade induced by FAS-ligand.	FAS Ligand	FAS Antigen	FAS-Associated Via Death Domain Protein (FADD)	Caspase-8	Caspase-3	----------	----------	----------
Pathway 1: Apoptosis cascade induced by Tumor Necrosis Factor-alpha (TNFα).	TNFα	TNFα Receptor	TNF Receptor-Associated Protein with a Death Domain (TRADD Protein)	Receptor-Interacting Serine-Threonine Kinase-1	TNF Receptor-Associated Factor-2 (TRAF2)	Serine-Threonine Protein Kinase-NIK	Conserved Helix-loop-helix Ubiquitous Kinase	I-Kappa-B-Alpha Protein
Aligned (Consensus) Pathway.	TNF Family Protein	TNF Receptor Family Protein	Adaptor Signaling Protein	Enzyme	Protein Organized by Function	----------	----------	----------

Table 3-4: Pair-wise pathway alignment – gap cost = -4.0.

Pathway Description	Protein Names							
Pathway 1: Apoptosis cascade induced by FAS-ligand.	FAS Ligand	FAS Antigen	FAS-Associated Via Death Domain Protein (FADD)	Caspase-8	----------	Caspase-3	----------	----------
Pathway 1: Apoptosis cascade induced by Tumor Necrosis Factor-alpha (TNFα).	TNFα	TNFα Receptor	TNF Receptor-Associated Protein with a Death Domain (TRADD Protein)	Receptor-Interacting Serine-Threonine Kinase-1	TNF Receptor-Associated Factor-2 (TRAF2)	Serine-Threonine Protein Kinase-NIK	Conserved Helix-loop-helix Ubiquitous Kinase	I-Kappa-B-Alpha Protein
Aligned (Consensus) Pathway.	TNF Family Protein	TNF Receptor Family Protein	Adaptor Signaling Protein	Enzyme	----------	Enzyme	----------	----------

Table 3-5: Pair-wise pathway alignment – gap cost = -2.0.

As can be seen from the above three tables (Table 3-3, Table 3-4 and Table 3-5), different gap cost would result in different alignments (i.e. consensus pathways). To determine which gap penalty would result in alignments that would be most meaningful biologically, one can use various criteria. For example, one could examine the protein classes in the consensus pathway – the one that consist of the largest number and the most specific protein classes represents the "true" alignment. All three consensus pathways consist of five protein classes. Thus, the number of protein classes cannot be used to rank the alignments. Using the criteria of maximizing the specificity of the protein classes, the second alignment (Table 3-4) is inferior because it has one "Protein Organized by Function", which is a generic protein. To distinguish between the first and third consensus pathways (Table 3-3 and Table 3-5), one could look at how the input proteins are aligned with each other. In the first alignment (Table 3-3), the "Adaptor Signaling Protein" is a result of aligning FADD and TRAF2. In the second alignment (Table 3-5), the "Adaptor Signaling Protein" is a result of aligning FADD and TRADD,

both of which has a death domain. Thus, one could say FADD and TRADD are more similar biologically than FADD and TRAF2[13]. Thus, the second alignment reflects a more biologically meaningful alignment.

From another point of view, the similarity matrix calculated from the NCI Thesaurus gives a mismatch a score of -1.0. Intuitively, a gap should cost more than a mismatch. Thus, a gap cost of -1.0×10^{-15} is clearly too small of a penalty for a gap.

3.4. Pair-wise Pathway Alignment

As mentioned above, since pathways can be represented by a string of protein names, one could do pair-wise pathways comparison in the same manner as comparing two strings. For example, two pathways could be aligned the same way as aligning two strings. The most straightforward way to align two strings is to use dynamic programming which guarantees an alignment with an optimal score.

3.4.1. Dynamic Programming

Please refer to [5] for a description of dynamic programming. As mentioned in section 3.3 (Similarity Matrix), the gap cost is calculated using the most basic model with cost-free end gaps: *cost of gap = constant * number of gaps*. The constant is arbitrarily chosen. The cost of a match, partial match or mismatch is determined by a similarity matrix function described in section 3.3 (Similarity Matrix).

---

[13] Using the NCI Thesaurus, both pairs of proteins ({FADD, TRADD} and {FADD, TRAF}) would have the same similarity score because they are all at the same level of protein classification under "Adaptor Signaling Protein".

3.4.2. Example Results

The following figure shows an example of a pair-wise pathway alignment result.

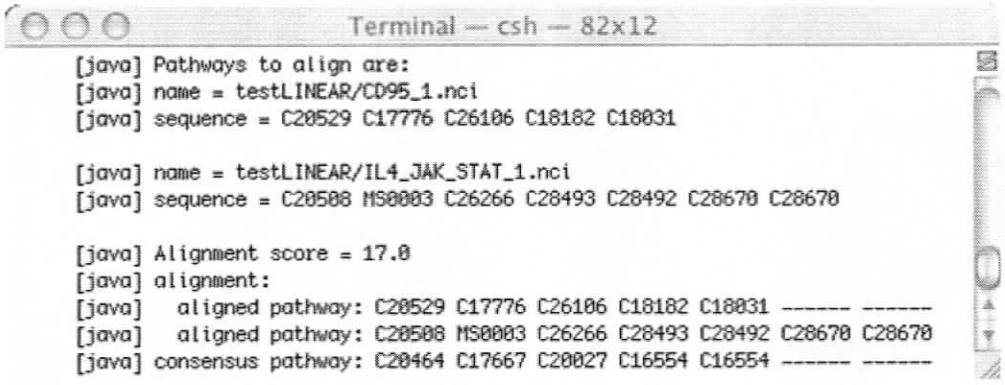


Figure 3-5: Example pair-wise pathway alignment result.

The following table shows the proteins names in the above pathway alignment example.

Pathway Description	Protein Names						
Pathway 1: Apoptosis cascade induced by FAS-ligand.	FAS Ligand	FAS Antigen	FAS-Associated Via Death Domain Protein (FADD)	Caspase-8	Caspase-3	----------	----------
Pathway 2: Interleukin-4 signaling pathway involving Jak-Stat.	Interleukin-4	Interleukin-4 Receptor Alpha	Interleukin-2 Receptor Gamma	Janus Kinase-1 (JAK-1)	Janus Kinase-3 (JAK-3)	Signal Transducer and Activator of Transcription-6 (STAT-6)	Signal Transducer and Activator of Transcription-6 (STAT-6)
Aligned (Consensus) Pathway.	Cytokine	Cytokine Receptor	Protein Organized by Function	Enzyme	Enzyme	----------	----------

Table 3-6: Pair-wise pathway alignment – protein names.

3.5. Clustering Based on Pair-wise Distance Score

This section describes cluster analysis on linear biochemical pathways (DFA without branching).

3.5.1. Overview of Clustering

Cluster analysis is a collection of statistical methods that is used to assign data points (literature in Statistics refer to them as "cases" – e.g. individual, things, or events) into groups (clusters). The assignment is based on some characteristics[14] of the data points so that data points in the same group are more similar with respect to these characteristics than data points from different groups. Thus, cluster analysis requires a notion of similarity i.e. a similarity scoring system. There are different types of cluster analysis.

Hierarchical vs. Non-hierarchical: Hierarchical cluster analysis involves grouping data such that the "resultant classification has an increasing number of nested classes" [53]. A phylogenetic tree is an example of a hierarchical cluster analysis. Non-hierarchical cluster analysis results in a classification that is one-level deep – i.e. the classification tree has only one internal node (the root node) and leave nodes. *Divisive vs. agglomerative:* These two categories differ from the way the cluster is built. Divisive clustering starts with all data points in one cluster and gradually partitions the data points into smaller clusters. Agglomerative clustering usually starts with each data point in a different cluster (i.e. starts with n clusters where n is the number of data points and each cluster contains one data point) and gradually merges the clusters into bigger clusters until one cluster is formed. *Monothetic or polythetic:* The distinction here deals with the number of characteristics of the data points that are used in assigning data points to the different clusters. Monothetic clustering uses only one characteristic while polythetic uses more than one characteristic. Some simple introductions to cluster analysis found on the Internet include [53] and [62]. Please refer to [11] for a more in-depth introduction.

---

[14] For example, if data points represent different dogs, the characteristics used in a cluster analysis may be the color of the fur, the length of the hair, the height, etc.

3.5.1.1. What is Cluster Analysis Used for

Cluster analysis has many interesting applications in bioinformatics. Since bioinformatics deals with large amounts of data, cluster analysis as well as other statistical tools often play significant roles. For example, cluster analysis is an integral part of many microarray experiments [8] [10] [31]. Microarray experiments are techniques allowing many characteristics of an individual's DNA or RNA sample to be determined in a single experiment. For example, an expression profile, a profile of the genes that are expressed, as indicated by the amount of mRNA in the individual's sample, is often constructed using microarray experiments. When expression profiles are available for a group of individuals, it is often helpful (insightful) to cluster the individuals based on their expression profiles and see if individuals that are in the same cluster have any physically observable (phenotypic) features – e.g. have a certain disease. These studies help scientists understand the molecular mechanism underlying a physical observable trait.

Another useful application of cluster analysis in bioinformatics is the construction of phylogenetic trees [9]. Phylogenetic tree is a useful tool to study the relationship between different species[15]. The relationship is based on how different species differ from each other on a number of characteristics. Species with more common characteristics would be considered more related than species with less common characteristics.

Consider a set of biochemical pathways. One could perform cluster analysis on them based on some of their characteristics. In this thesis, the characteristics of a biochemical

---

[15] The author believes that phylogenetic tree is a tool that shows design relationships (roughly similar to class inheritance relationship in object-oriented programming). However, literatures in biology suggest that evolutionary relationships are depicted by phylogenetic trees.

pathway are captured in its representation as a state machine. Cluster analysis based on these characteristics would enable one to build a phylogenetic tree of biochemical pathways. For the biochemical pathways analysis in this thesis, pathways within the same species are considered. Therefore, it might not be appropriate to infer any evolutionary relationship. Cluster analysis can be used as a classification tool to classify biochemical pathways. It would be interesting to see if pathways that are classified (clustered) together have any significant biological meaning. For example, they may have similar biological function or they may have similar behavior using similar control logics.

A less prominent yet significant application of cluster analysis can be found in multiple sequence alignment. Not all multiple sequence alignment methods use clustering methods. A class of methods known as progressive alignment methods uses clustering methods – these are some of the fastest and most commonly used multiple sequence alignment methods. Please refer to section 3.7.3 (Multiple Pathway Alignment) for a more detail explanation on progressive alignment methods.

3.5.2. UPGMA

Since the focus of the pathway comparison done in this thesis is not to discover evolutionary relationships, a simple clustering method can be used to cluster biochemical pathways based on their biological function and behavior – one that does not adjust the cluster for evolutionary "correctness". The unweighted pair group method using arithmetic averages, UPGMA, is one of the least complicated clustering methods.

As an aside, a number of clustering algorithms have been developed to deal with issues in construction of phylogenetic trees. For example, Fitch-Margoliash method (used in

Feng-Doolittle multiple sequence alignment method [12]) and neighbour-joining method (used in Clustalw multiple sequence alignment method [44]) are two methods that do not make the "molecular clock" [9] assumption, which UPGMA does make. The "molecular clock" assumption says that all species diverge at the same rate and that in the "true" phylogenetic tree, the branch length can be used to represent time. However, when constructing a phylogenetic tree, only data on changes (edit distance) are available. The following two trees show how the "molecular clock" assumption can be incorrect. This diagram is taken from [51].

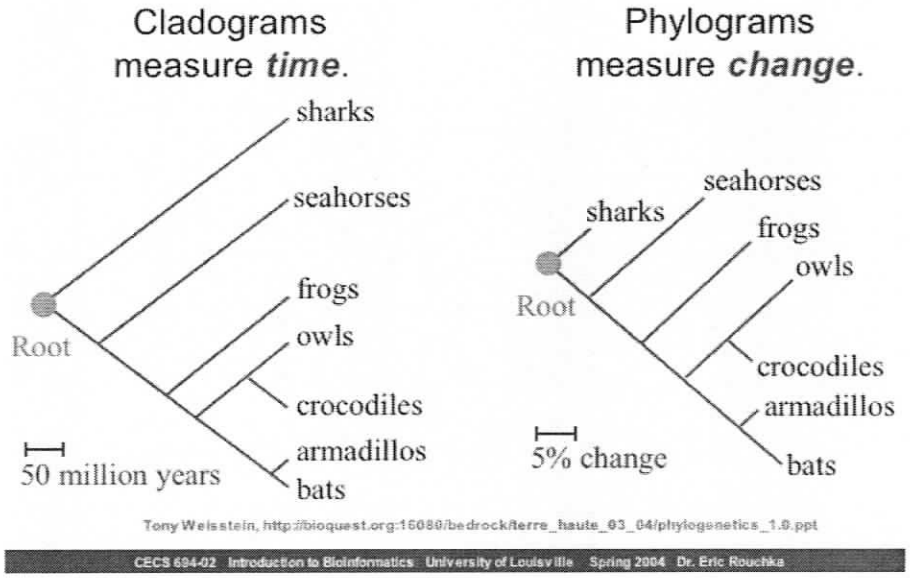


Figure 3-6: Two trees showing evolutionary time and edit distance (changes).

3.5.2.1. Distance Score Calculation

The pair-wise comparison using dynamic programming described in section 3.4 (Pair-wise Pathway Alignment) allows the calculation of a similarity score between two pathways. For cluster analysis, a distance score is needed. The distance score calculation method used by Feng & Doolittle [12] in their multiple sequence alignment algorithm was used.

$$D = -\log S_{eff} = -\log \frac{S_{obs} - S_{rand}}{S_{\max} - S_{rand}}$$

Equation 4: Feng & Doolittle's distance score calculation method.

The above equation is taken from [9]. D is the distance score. S_{obs} is the similarity score from pair-wise pathway alignment using dynamic programming. S_{rand} is the similarity score between two randomly generated sequences. Randomly generated sequence is generated by picking a sequence of terms (protein or protein class names) from the NCI Thesaurus. The S_{rand} is generated as follows. For each sequence length from 1 to 20, pair-wise alignments using dynamic programming were done for 100 pairs of randomly generated, equal length sequences. The results were plotted and a best-fit polynomial (2-degree) was generated.

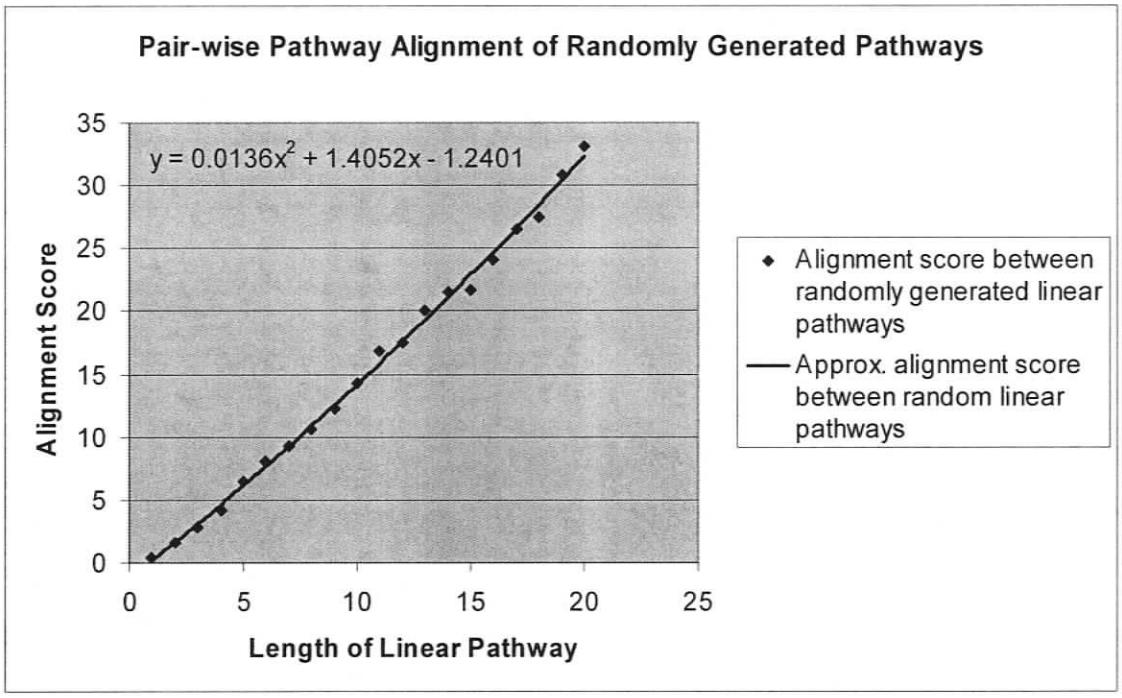


Figure 3-7: Pair-wise pathway alignment of randomly generated pathways.

For a pair of sequences of unequal length, S_{rand} is calculated by "entering in" the average length of the sequences into the equation shown in the above figure. For example, for a pair of sequence of length 7 and 9:

$$S_{rand} = 0.0136(\frac{7+9}{2}) + 1.4052(\frac{7+9}{2}) + 1.2401 = 10.57$$

It is possible for S_{rand} to be smaller than $S_{obs,}$ in which case, Equation 4 would not return a valid number because of the log of a negative number. In this case, the value of S_{obs} would be set to:

$$S_{obs} = S_{rand} + \varepsilon$$

Where, ε is an arbitrarily set constant that represents a very small number.

S_{max} is the alignment score of a pair of identical sequence. For a pair of non-identical sequences, S_{max} is taken to be the average alignment scores of the pair-wise alignment of the first sequence against itself, and the second sequence against itself.

3.5.2.2. UPGMA algorithm

The following is a description of the UPGMA algorithm taken from [9]. Let d_{ij}, the distance between two clusters C_i and C_j, be defined as follows.

$$d_{ij} = \frac{1}{|C_i| \cdot |C_j|} \cdot \sum_{p \in C_i, q \in C_j} d_{pq},$$

Equation 5: Calculation of d_{ij}

where $|C_i|$ and $|C_j|$ denote the number of sequences in clusters i and j, respectively. If C_k is the union of the two clusters C_i and C_j, and if C_l is any other cluster, then one could calculate d_{kl} as follows.

$$d_{kl} = \frac{d_{il} \cdot |C_i| + d_{jl} \cdot |C_j|}{|C_i| + |C_j|}$$

Equation 6: Calculation of d_{kl}.

Initialization:

- Assign each sequence *i* to its own cluster C_i,
- Define one leaf of *T* (the cluster analysis tree) for each sequence, and place at height zero.

Iteration:

- Determine the two clusters *i*, *j* for which d_{ij} is minimal. (If there are several equidistant minimal pairs, pick one randomly.)
- Define a new cluster *k* with $C_k = C_i \cup C_j$, and define d_{kl} for all *l* by Equation 6: Calculation of d_{kl}.
- Define a node *k* with daughter nodes *i* and *j*, and place it at height $d_{ij}/2$.
- Add *k* to the current clusters and remove *i* and *j*.

Termination:

- When only two clusters *i*, *j* remain, place the root at height $d_{ij}/2$.

Please refer to section 3.7 (Example Results) for an example cluster analysis of biochemical pathways.

3.6. Multiple Pathway Alignment

This section describes multiple pathway alignment analysis on biochemical pathways expressed as a sequence of protein names.

3.6.1. Introduction – What is It and Why Use It

This section introduces multiple sequence alignment. It discusses how multiple sequence alignment methods can used in biochemical pathway alignment and describes some potential benefits for doing multiple pathway alignment.

3.6.1.1. Methods Used for Multiple Sequence Alignment

This section describes some multiple sequence alignment methods commonly used in aligning biological sequences such as DNA and amino acid sequences.

3.6.1.1.1. N-Dimensional Dynamic Programming

This method gives a mathematically optimal alignment. However, it is computationally expensive. For example, aligning n sequences, where each sequence has length m, would require $O(m^n)$ memory space and $O(2^n m^{n)}$ time [9].

3.6.1.1.2. Progressive Alignment

Progressive alignment methods are some of the most commonly used multiple sequence alignment methods. This is because these methods are fast and the alignments they produce are satisfactory. The basic algorithm is as follows.

1. For each pair of sequences, do a pair-wise alignment. From these alignment results (similarity scores), calculate the corresponding distance scores (i.e. distance between sequences) and construct a distance matrix.
2. From the distance matrix (constructed in step 1), construct a "guide tree" using some clustering method (e.g. Neighbor-joining for Clustalw).

3. Starting from the leaves of the guide tree, progressively align the sequences starting with the pair of sequences that have the smallest distance between them. Doing so, one would need to be able to construct three types of alignments:

   - sequence with sequence
   - sequence with alignment
   - alignment with alignment

   When doing alignment with alignments, the alignments that are being aligned, either against a sequence or another alignment, stays intact. For example, consider aligning sequence *C* with alignment *D*, the alignment between sequence *A* and *B*. If one needs to insert a gap to alignment *D*, one would need to insert a gap to both sequence *A* and *B* at the same position. One cannot insert a gap only to sequence *B* and modify the existing alignment between sequence *A* and *B*.

The rationale behind this algorithm is that alignments between closely related (i.e. more similar) sequences are more reliable than alignments between more distant related sequences. One of the major drawbacks of progressive alignment methods is: "once a gap, always a gap". That is, if the alignments early on in the progressive alignment process are "wrong", these "wrong alignments" propagate through the whole multiple sequence alignment process.

3.6.1.2. Feng and Doolittle Progressive Alignment

This section describes the progressive alignment method developed by Feng and Doolittle [12], one of the earliest and most simple progressive alignment methods. The following is a description of this method taken from [9].

1. Calculate a diagonal matrix of $N(N-1)/2$ distances between all pairs of N sequences by standard pair-wise alignment by converting raw alignment scores to approximate pair-wise "distances". Please refer to section 3.5.2.1 (Distance Score Calculation) for a description of how the distance scores are calculated.
2. Construct a guide tree from the distance matrix using the clustering algorithm by Fitch & Margoliash [13].
3. Starting from the first node added to the tree, align the child nodes, which may be two sequences, a sequence and an alignment, or two alignments. Repeat for all other nodes in the order that they were added to the tree (i.e. from most similar pairs to least similar pairs) until all sequences have been aligned.

3.7. Example Results

The above-described pathway analysis methods were applied to six signaling pathways. This section describes the results of cluster analysis and multiple pathway alignment.

3.7.1. Input Pathways

This section briefly describes the input pathways.

3.7.1.1. Overview of Apoptosis

Apoptosis, the regulated destruction of a cell also known as programmed cell death, is an important process for multicellular organisms to destroy cells that are in excess, in the way or potentially dangerous [21]. For example, apoptosis is important during embryogenesis (the development and growth of the organism), cellular homeostasis (maintenance of the body cell counts), tissue atrophy (loss of cells in tissues), and tumor

regression. In addition, in the immune system, apoptosis helps destroy self-reactive **T** and **B cells** [19]. The decision for a cell to commit to apoptosis cannot be taken lightly. Therefore, a complex control system has been observed in cells from different organisms to regulate apoptosis. There are two major signaling pathways of apoptosis, the death receptor pathway and the mitochondrial pathway [19]. In this thesis, only the death receptor pathway is studied. Briefly, in the death receptor pathway, ligands (e.g. cytokine) bind to the "death receptor" which causes activation of other proteins, which activate other proteins. The signals cascade through these protein activations until the molecular machineries, which carry out the specific tasks for apoptosis, have been activated.

Caspases, a class of enzymes that is currently shown to cause most of the visible characteristics of apoptotic cell death, can be thought of as the "central executioners of the apoptotic pathway" [21]. In fact, eliminating caspases activity slows down or even prevents apoptosis [21]. Caspases, themselves, do not do anything to "kill" the cell. Instead, they activate other proteins (Caspase substrates) that carry out the "killing". For example, one such protein is the caspase-activated DNase (CAD), which is responsible for cutting genomic DNA. In this thesis, the signaling pathway from the activation of the "death receptor" to the activation of caspases is studied. The following is a diagram of the signaling pathway of apoptosis from KEGG [59]. Please refer to Appendix B. "KEGG Diagram Legend" for the legend of this figure.

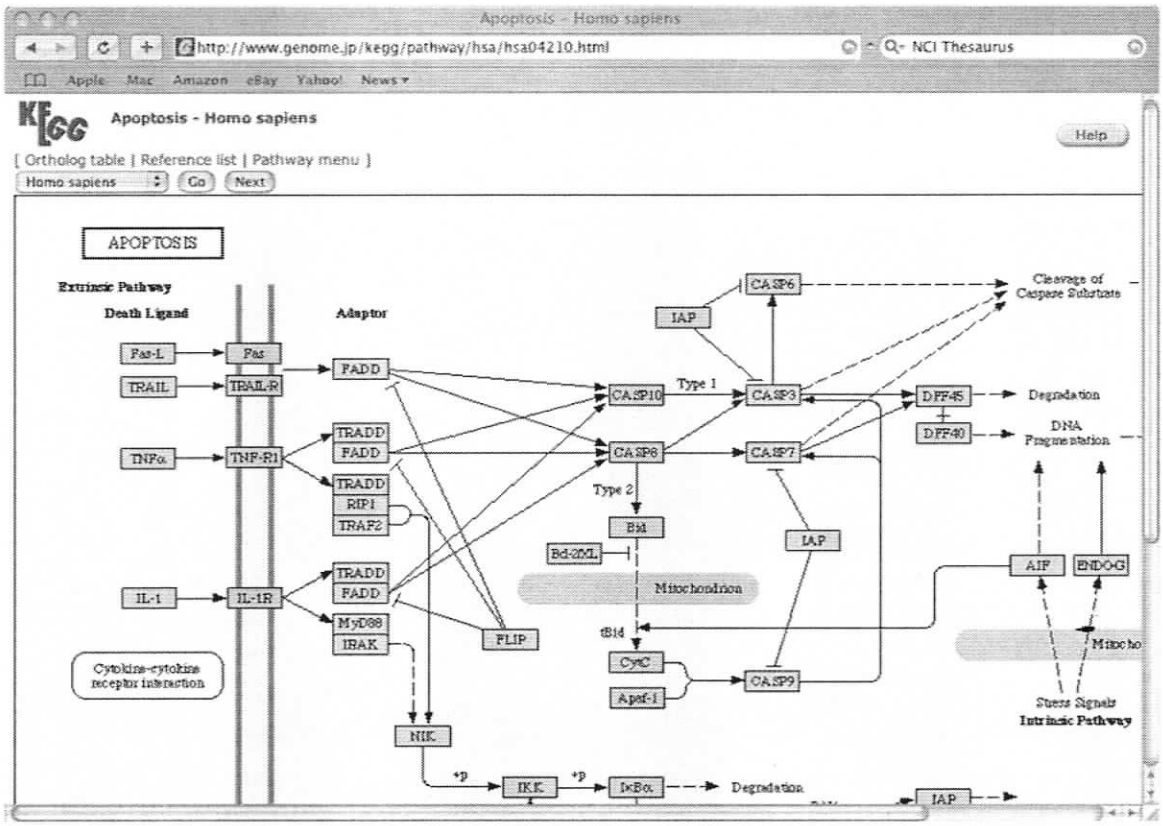


Figure 3-8: Signaling pathway of apoptosis from KEGG.

In the above diagram (Figure 3-8), the two vertical lines on the left represent the cell membrane (cell membrane is a lipid bilayer).

3.7.1.2. Apoptosis Signaled by Fas (CD95) Ligand

From the KEGG diagram (Figure 3-8), one could extract the following linear pathways initiated by the binding of Fas-ligand (Fas-L). `FADD` is an acronym for `Fas-Associated_Via_Death_Domain_Protein`.

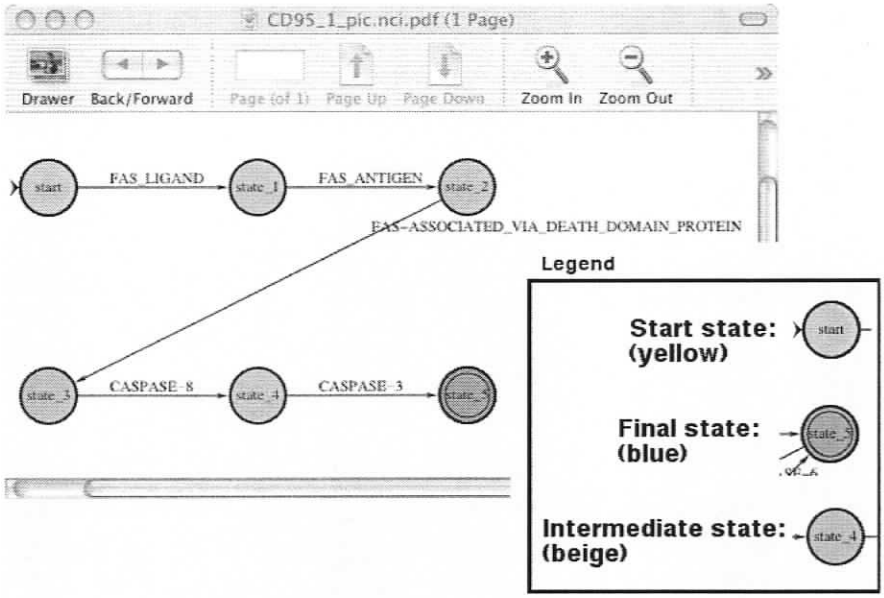


Figure 3-9: Fas-ligand death receptor signaling pathway 1.

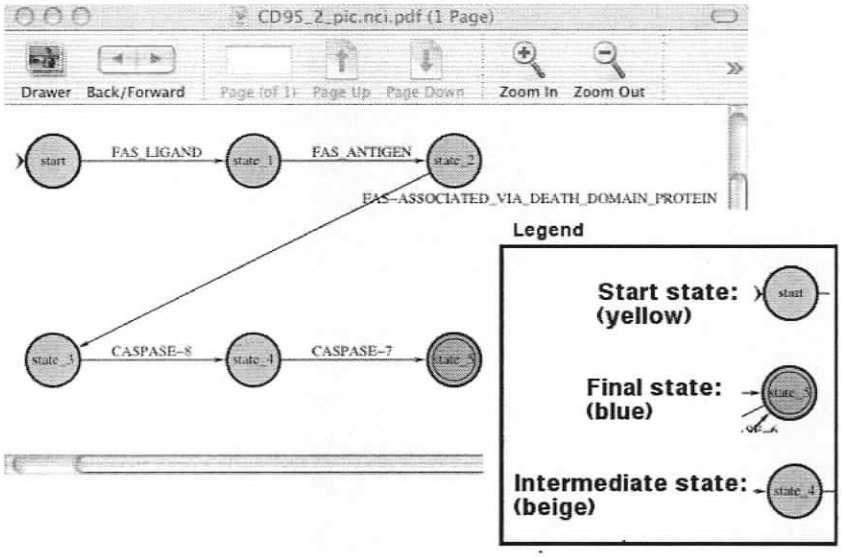


Figure 3-10: Fas-ligand death receptor signaling pathway 2.

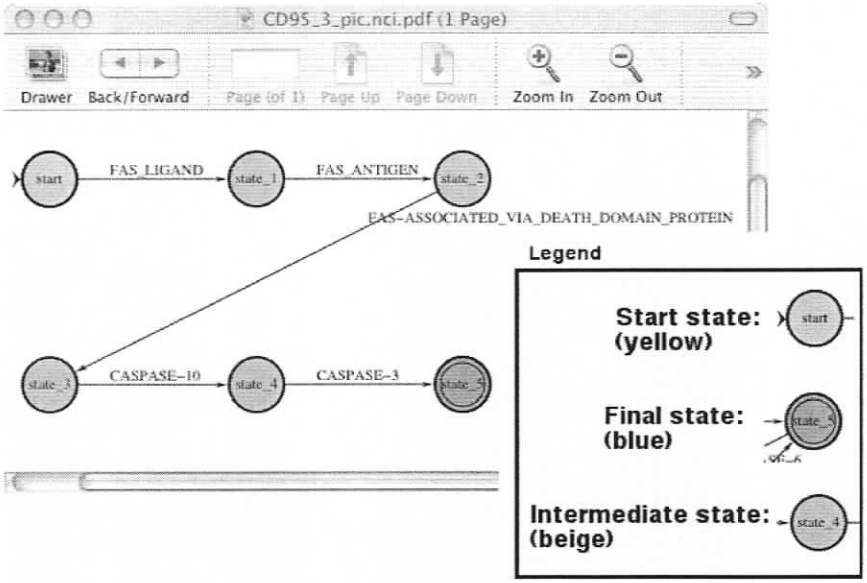


Figure 3-11: Fas-ligand death receptor signaling pathway 3.

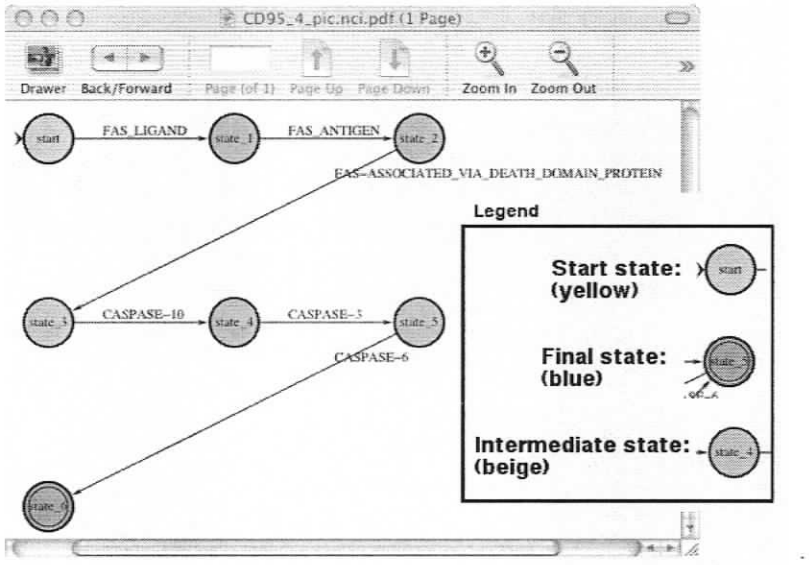


Figure 3-12: Fas-ligand death receptor signaling pathway 4.

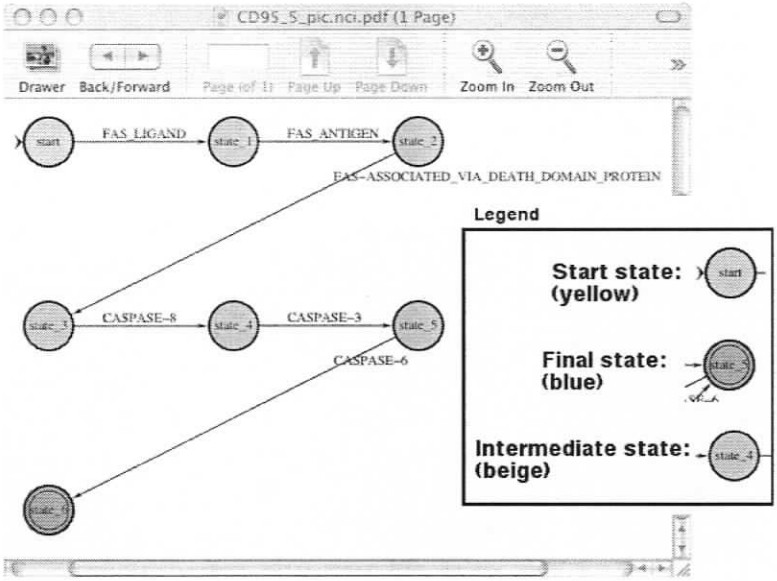


Figure 3-13: Fas-ligand death receptor signaling pathway 5.

The current understanding of the death-receptor pathway is that the binding of fas-ligand to the receptor causes the receptors to cluster together to form a "death-inducing signaling complex". Multiple non-active capase-8's (procaspase-8) then bind to this complex via the adaptor molecule, FADD. When a number of procaspase-8's are close to each other (because they are bound to the same death-receptor complex), they activate each other. This process is called "induced proximity" [21]. This aspect of the pathway ("induced proximity") is not captured in the DFA model shown in the above figures: Figure 3-9, Figure 3-10, Figure 3-11, Figure 3-12, and Figure 3-13.

All of the above DFA's represent only one string. Therefore, one could represent these linear pathways as a sequence of protein names.

FAS_L → FAS → FADD → Caspase-8 → Caspase-3
FAS_L → FAS → FADD → Caspase-8 → Caspase-3 → Caspase-6
FAS_L → FAS → FADD → Caspase-8 → Caspase-7
FAS_L → FAS → FADD → Caspase-10 → Caspase-3
FAS_L → FAS → FADD → Caspase-10 → Caspase-3 → Caspase-6

Table 3-7: Fas-ligand signaling pathway represented as strings of protein names.

3.7.1.3. Apoptosis Signaled by TNF-Alpha

From the KEGG diagram (Figure 3-8), one could extract the following linear pathways initiated by the binding of TNFα, represented as a sequence of protein names.

TNFα → TNFα-receptor → TRADD → RIP1 → TRAF2 → NIK → IKK → IκBα
TNFα → TNFα-receptor → TRADD → FADD → Caspase-10 → Caspase-3
TNFα → TNFα-receptor → TRADD → FADD → Caspase-10 → Caspase-3 → Caspase-6
TNFα → TNFα-receptor → TRADD → FADD → Caspase-8 → Caspase-3
TNFα → TNFα-receptor → TRADD → FADD → Caspase-8 → Caspase-3 → Caspase-6
TNFα → TNFα-receptor → TRADD → FADD → Caspase-8 → Caspase-7

Table 3-8: TNFα signaling pathway represented as strings of protein names.

In addition to apoptosis, TNFα also induces inflammatory response via the first pathway in the above table (Table 3-8).

3.7.1.4. Type I Interferon (Alpha/Beta IFN) Pathway

Interferons (IFNs) are glycoprotein **cytokines** produced and secreted by certain cells that induce antiviral state in other cells. They help regulate the immune response. Interferon alpha (IFNα, also known as type 1 interferon, leukocyte interferon, and lymphoblast interferon) induces resistance to viruses and inhibits cell proliferation. It regulates expression of class I MHC[16] molecules on nucleated cells. Interferon beta (IFNβ, also known as type 1 interferon and fibroblast interferon) has similar function. The following is a diagram of the IFNα/β pathway from BioCarta [54]. Please refer to Appendix C. "BioCarta Diagram Legend" for the legend of this figure.

---

[16] In terms of biological function, class I MHC is responsible for presenting endogenous antigen (antigen that are produced inside the host cell – e.g. viral protein synthesized in the virus-infected cell) while class II MHC is responsible for presenting exogenous antigen (antigen that are produced outside the cell – e.g. bacteria) [17].

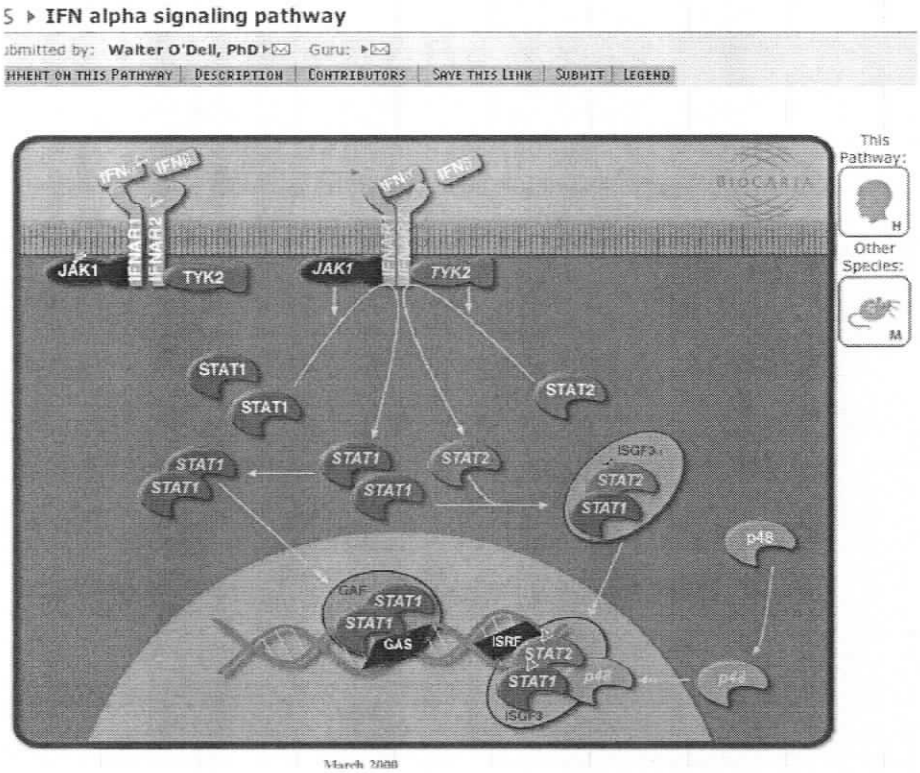


Figure 3-14: IFN α/β signaling pathway from BioCarta.

IFNα binds to the receptor. This activates the receptor-associated enzymes (JAK1 and Tyrosine Kinase 2 (Tyk2)). These enzymes activate STAT1 and STAT2, which causes them to heterodimerize (bind together). The STAT1-STAT2 compound then translocates to the nucleus and associate with p48 protein, also known as interferon regulatory factor 9 (IRF9), to form the ISGF3 complex (IFNα-stimulated gene response factor), which is a transcription factor for activating transcription of target genes [63]. One could extract a linear pathway from the above pathway description.

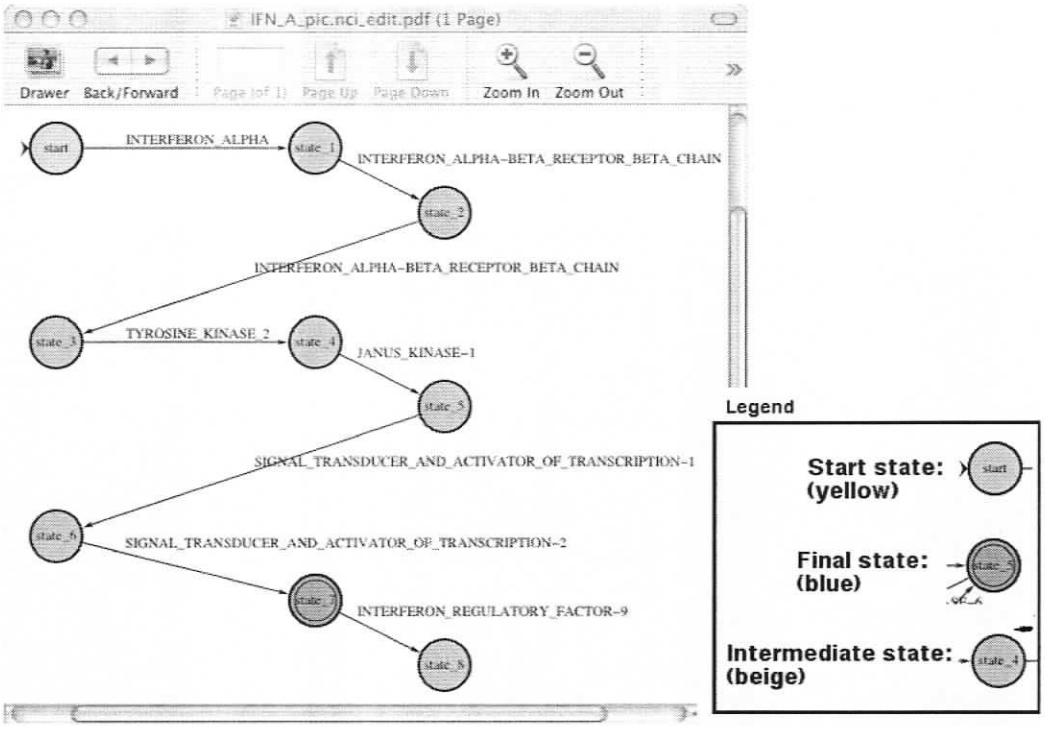


Figure 3-15: Interferon α signaling pathway represented as DFA without branching.

The IFNβ pathway could be represented with the same DFA as the above with IFNα replaced by IFNβ.

The state machines of IFNα and IFNβ pathways represent only one string.

IFNα pathway:

IFNα → IFNαR1 → IFNαR2 → Tyk2 → JAK1 → STAT1 → STAT2 → IRF9

IFNβ pathway:

IFNβ → IFNαR1 → IFNαR2 → Tyk2 → JAK1 → STAT1 → STAT2 → IRF9

3.7.1.5. Type II Interferon (Gamma) Pathway

Interferon gamma (IFNγ, also known as Type 2 interferon, immune interferon, macrophage-activating factor (MAF), and T cell interferon) affects activation, growth, and differentiation of **T cells**, **B cells**, and macrophages. It up-regulates MHC expression

in antigen-presenting cells. It is a signature cytokine of T_H1[17] differentiation. It induces weak anti-viral and anti-proliferative activities [17]. Interferon gamma uses a **JAK-STAT** signaling pathway, a pathway that is used by other cytokines also [25]. The following is a diagram of the IFNγ signaling pathway from BioCarta [54]. Please refer to Appendix C. "BioCarta Diagram Legend" for the legend of this figure.

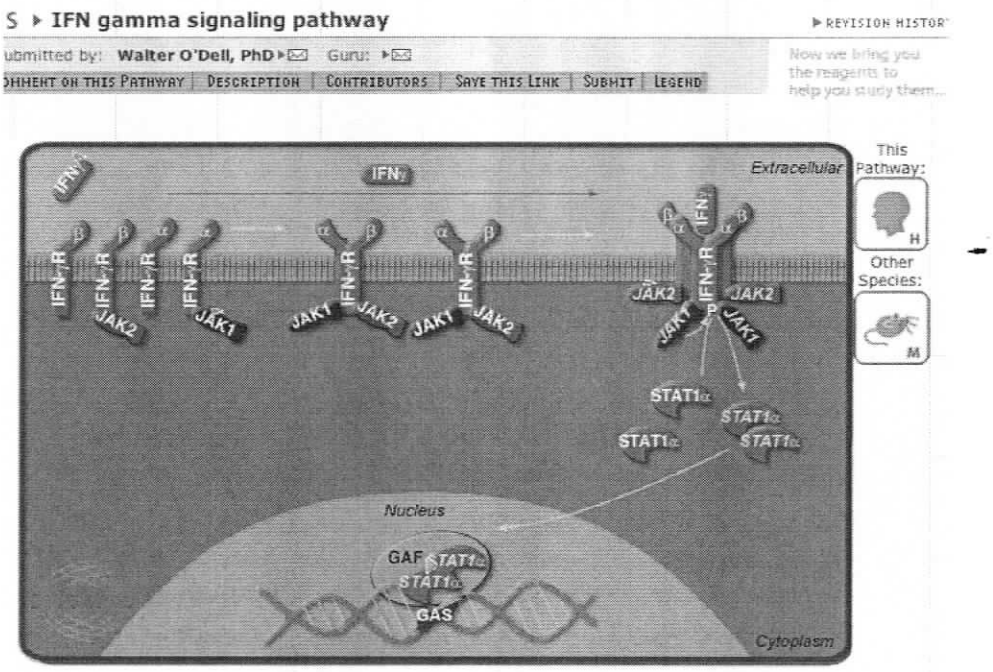


Figure 3-16: IFN γ signaling pathway from BioCarta.

Binding of IFNγ to the IFNγ receptor stimulates the activation of two receptor associated enzymes: JAK1 and JAK2. This leads to the phosphorylation of an amino acid (tyrosine located at position 440) residue of the IFNγ receptor. This phosphorylated amino acid is recognized by the SH2 domain of STAT1, thus attracting STAT1 to bind to the IFNγ receptor. Once at the receptor, STAT1 is activated by a single phosphorylation event,

[17] T_H1 response produces a cytokine profile that supports inflammation and cell mediated responses [17].

whereupon it homodimerizes (i.e. two STAT1 protein binds together), translocates to the nucleus, and promotes the induction of GAS (γ-activated sequence) driven target genes [42]. Please refer to the following references for more detail information of the IFNγ signaling pathway: [20] and [47].

From the above description, the following linear path of interacting proteins (linear pathway) can be extracted. Please note, concurrent events are treated as arbitrary interleaving of sequential events – e.g. the activation of the two STAT1 proteins.

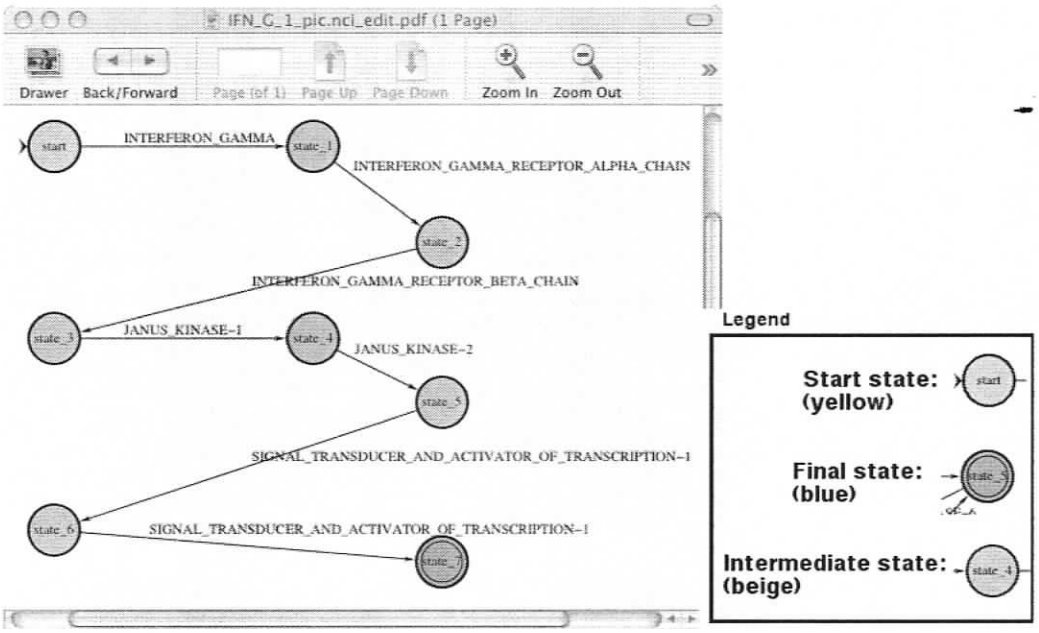


Figure 3-17: Interferon γ signaling pathway represented as DFA without branching.

The above state machine represents only one string, a sequence of protein names:

IFNγ → IFNγRα → IFNγRβ → JAK1 → JAK2 → STAT1 → STAT1

3.7.1.6. Interleukin 4 (IL-4) Pathway

IL-4 (also known as B cell-stimulating factor 1, BSF-1) is a cytokine, which promotes growth and development of various types of white blood cells (e.g. T and B cells) as well as certain cell types outside the immune system such as endothelial cells and fibroblast. It

is produced by T_H2 cells, mast cells and basophils. The signaling pathway of IL-4 uses the JAK-STAT proteins [17]. IL-4 also causes some responses, which are associated with allergy and asthma [29]. The following is a diagram of the IL-4 signaling pathway from BioCarta [54]. Please refer to Appendix C. "BioCarta Diagram Legend" for the legend of this figure.

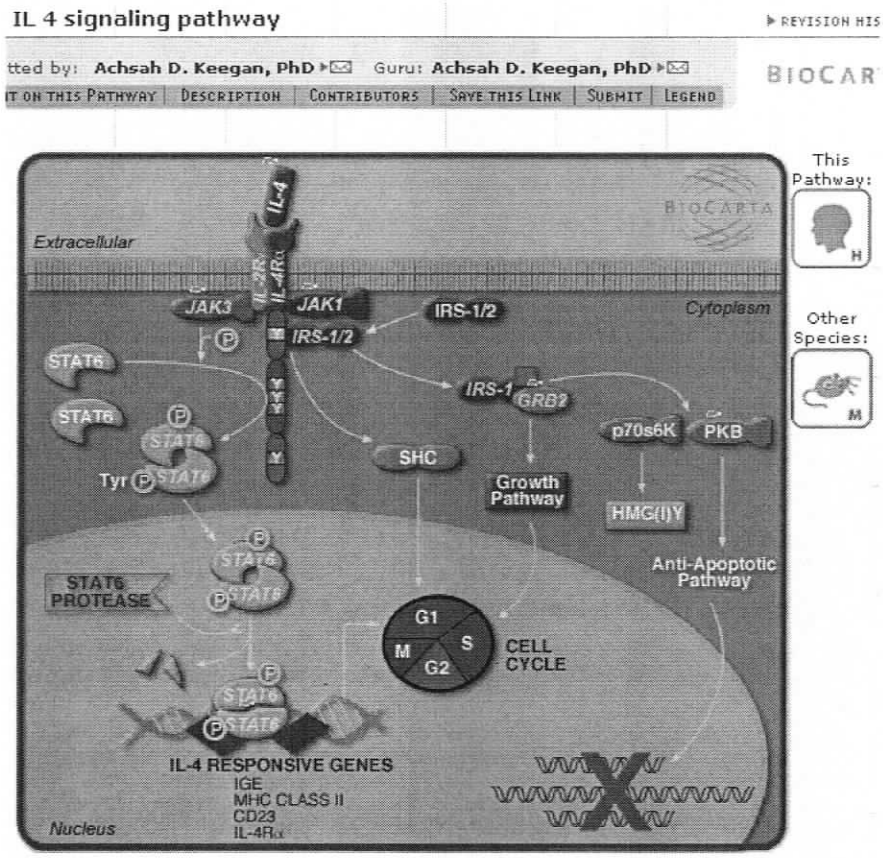


Figure 3-18: IL-4 signaling pathway from BioCarta.

There are two major signaling pathways activated by IL-4. One leads to the activation of STAT6 (a latent cytoplasmic transcription factor), and the other leads to the activation of insulin receptor substrate (IRS), which leads to the activation of phosphoinositide 3-kinase (PI3K) pathway, which deals with growth, survival and regulation of gene expression. This thesis focuses only on the former pathway (activation of STAT6).

The IL-4 signaling pathway differs depending on whether the cell is of hematopoietic lineage (lineage of blood cells) or not. In hematopoietic lineage cell, binding of IL-4 to IL-4 receptor alpha (IL-4Rα) chain causes IL-4Rα to bind to a protein called common gamma chain (γC). The resulting complex is referred to as type I receptor. In nonhematopoietic lineage cells, a type II receptor is formed instead. In type II receptor, IL-4Rα binds to IL-13 receptor alpha (IL-13Rα1) instead of γC. The receptor associates with JAK enzymes. IL-4Rα associates with JAK1, γC associates with JAK3, and IL-13Rα1 associates with Tyk2 or JAK2. These JAK enzymes then activate STAT6, which regulates gene expressions. The following is an example of the function of STAT6: "STAT6 is central in gene regulation and the IL-4- and IL-13-regulated allergic responses, including T_H2 differentiation, IgE (immunoglobulin E) production, and chemokine and mucus production at sites of allergic inflammation" [29]. From the above diagram (Figure 3-18) and the above description, one could extract the following linear pathways, which leads to the activation of STAT6.

IL-4 → IL-4Rα → γC → JAK1 → JAK3 → STAT6 → STAT6
IL-4 → IL-4Rα → IL-13Rα1 → JAK1 → JAK2 → TYK2 → STAT6 → STAT6

Table 3-9: IL-4 signaling pathway represented as strings of protein names.

Please note, when STAT6 is activated by JAK, it dimerizes with another STAT6 to form the active transcription factor. This is the reason for having two STAT6's in the above linear pathways.

3.7.1.7. Interleukin 13 (IL-13) Pathway

The signaling pathway of IL-13 is similar to the one of IL-4. Binding of IL-13 leads to the activation of STAT6, which leads to expression of IL-13 regulated genes. From the above diagram (Figure 3-18), one could extract the following linear pathway.

IL-13 → IL-13Rα1 → IL-4Rα → JAK1 → JAK2 → Tyk2 → STAT6 → STAT6

3.7.2. Clustering

The following diagram shows the result of clustering all the linear pathways described in the previous section (3.7.1 Input Pathways). The dendrogram was generated using R [61].

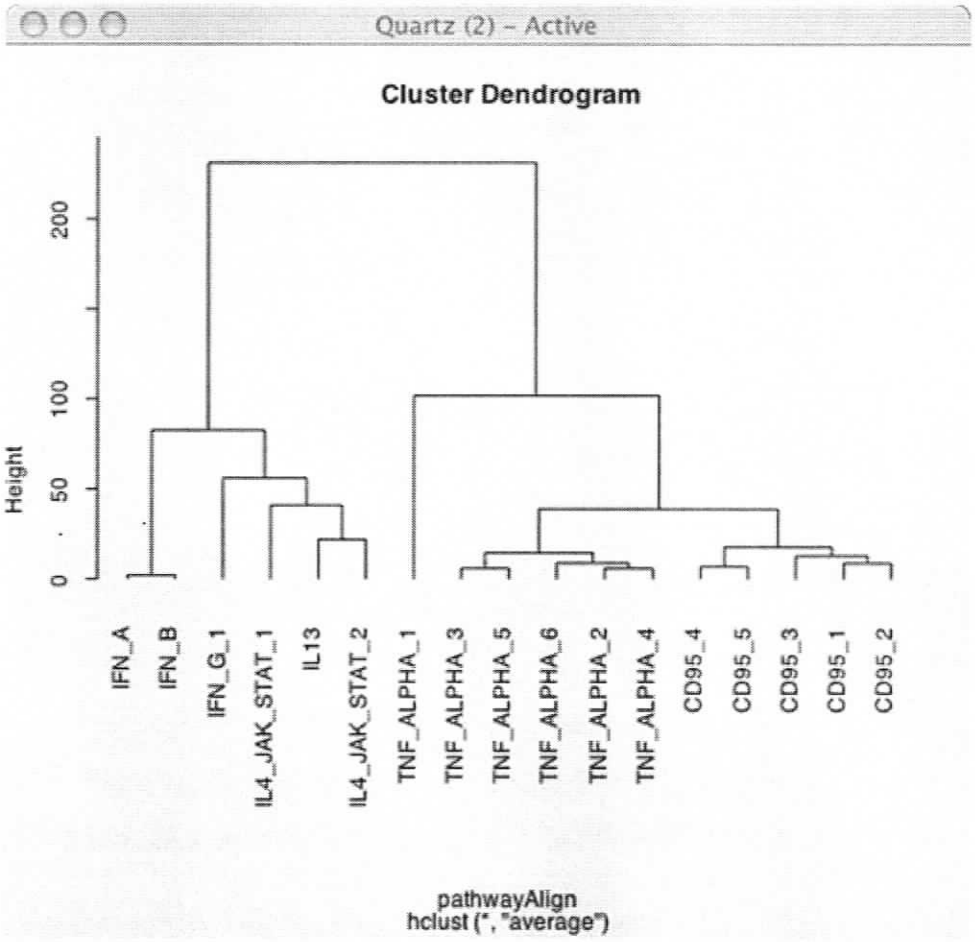


Figure 3-19: UPGMA clustering of linear pathways.

As mentioned before, the clustering is done using UPGMA and the pair-wise pathway comparison used to generate the distance matrix is done using dynamic programming (with no end-gap penalty). The two major clusters generated reflect the biological function of the pathways. The first major cluster includes pathways that are responsible for immune response (e.g. inflammation). These include the interferon and interleukin pathways. The second major cluster includes pathways involved in apoptosis – the TNF and CD95 pathways. One TNF pathway, `TNF_ALPHA_1`, stands out by itself. This is because this pathway is involved in the inflammatory response via IκBα, not apoptosis. If one were to examine the individual linear pathways, one could see that the clustering results are actually quite trivial because pathways that are clustered together often share the same proteins. However, as a preliminary result, this clustering experiment shows that it is possible to cluster linear pathways into clusters that share similar biological function.

3.7.3. Multiple Pathway Alignment

It would not yield any interesting result if one aligned all the above pathways in one multiple sequence alignment as Figure 3-20 shows. In this diagram, proteins are represented by NCI protein code for display purposes. Please refer to appendix A, "NCI Thesaurus Protein Code/Name Table", for the common protein names.

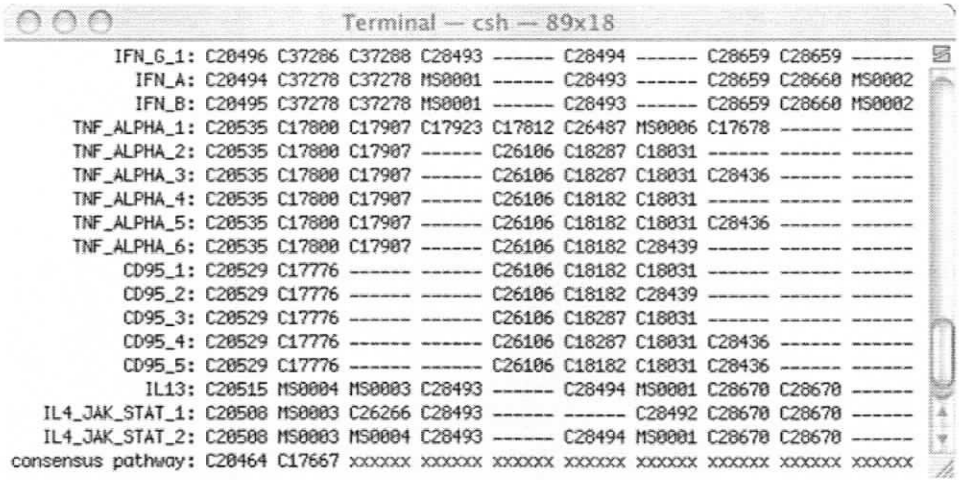


Figure 3-20: Multiple pathway alignment of 17 linear pathways.

The consensus pathway (the common features among all linear pathways) is not very interesting: `Cytokine` → `Cytokine_Receptor`. The following two diagrams show the multiple pathway alignment of the pathways in the two major clusters found in the clustering experiment described in section 3.7.2.

Figure 3-21: Multiple pathway alignment of the signaling pathways (linear pathways) of IFNγ, IFNα, IFNβ, IL-13, and IL-4.

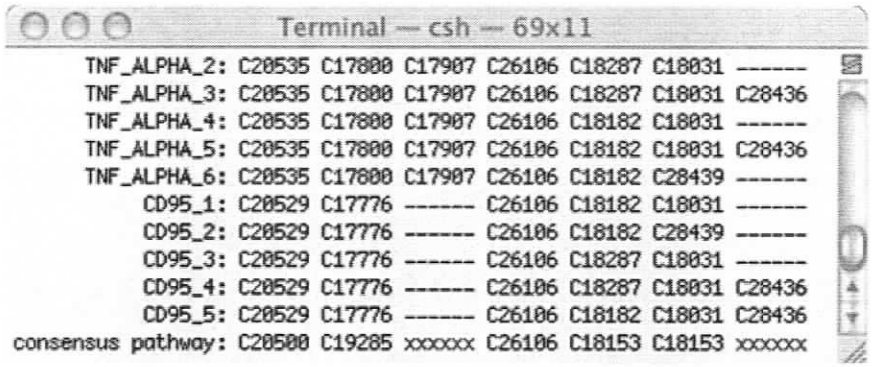


Figure 3-22: Multiple pathway alignment of the signaling pathways (linear pathways) of TNFα and CD95.

The consensus pathway between the signaling pathways of IFNγ, IFNα, IFNβ, IL-13, and IL-4 is as follows: `Cytokine` → `Cytokine_Receptor` → `Cell_Surface_Receptor` → `Protein_Tyrosine_Kinase` → ••• → `Signal_Transducer_and_Activator_of_Transcription` → `Signal_Transducer_and_Activator_of_Transcription`

The consensus pathway between the signaling pathways of TNFα and CD95 is as follows: `Tumor_Necrosis_Factor_Family_Protein` → `TNF_Receptor_Family_Protein` → ••• → `Fas-Associated_Via_Death_Domain_Protein` → `Caspase` → `Caspase`

From the above two consensus pathways, it can be seen that one major difference between the immune response signaling pathway (IFNγ, IFNα, IFNβ, IL-13, and IL-4) and the apoptosis signaling pathway (TNFα and CD95) is that the downstream "executioner" protein of the immune response pathway is a transcription factor while that of the apoptosis pathway is an enzyme. This observation is only valid for the pathways examined above. More pathways are needed to be examined to make a more general claim.

3.8. Mini-Conclusion

The contributions made in the research described in chapter 3 are as follows.

- A pathway representation model was developed using strings of protein names – linear pathways.

- Pair wise alignments of linear pathways were done using dynamic programming. The similarity matrix, assessing the similarity between a pair of proteins, was calculated using the NCI Thesaurus.
- UPGMA was used to cluster 17 linear pathways. The clustering algorithm was able to classify the linear pathways into the correct biological function groups.
- A progressive multiple sequence alignment algorithm was used to perform multiple linear pathways alignment.

Chapter 4

4. Part II: Abstraction to DFA without Cycles

This chapter describes the abstraction of biochemical pathways using deterministic finite automata (DFA).

4.1. From Pathways to DFA without Cycles

In chapter 3, a string of protein names was used as an abstraction of biochemical pathways (linear pathways). One limitation of this model is that in each state, only one input symbol is valid and hence the fact that a biochemical system can react to the introduction of different chemical species is ignored. This model can be extended slightly to capture this aspect of biochemical pathway. A biochemical pathway can be represented as a deterministic finite automaton (DFA) using the same abstraction principle used in chapter 3 for linear pathway. For simplicity, this thesis only analyzes DFA that has no "loops" i.e.

$$q_i \xrightarrow{a_0} \ldots \not\rightarrow \ldots \xrightarrow{a_n} q_i$$

where, q_i represents state i, and a_i represents an input symbol to state i. A special property of this kind of automata is that the number of strings it represents is always finite – a property that is needed for the DFA comparison algorithm described later in this thesis. Nondeterministic finite automata are not used in this thesis to represent biochemical pathways because it would imply that an introduction of a chemical species to a biochemical system would result in different outcomes, each outcome with some probability. Although this is true for molecular interactions at the atomic level (e.g.

whether a collision between two molecules results in a chemical reaction can be expressed as a probability function), at a higher level (cellular level), this is not true. This is because, at the cellular level, one would be observing the average behavior of a large population of chemicals – i.e. although the fate of individual atoms is not deterministic, the fate of the whole population of chemical species is deterministic.

4.2. Pair-wise Pathway Alignment

Pair-wise alignment of pathways represented as DFAs involves the following steps:

1. Find all the strings represented by the two DFA.
2. For each string in DFA 1, do pair-wise alignment between all strings in DFA 2. Keep the best alignment. Do the same with all the strings in DFA 2, aligned against the strings in DFA 1. At the end, m + n alignments will result, where m is the number of strings represented by DFA 1, and n is the number of strings represented by DFA 2.
3. Take the alignments from step 2 to construct a "consensus DFA".

The rationale for the alignment algorithm described is as follows. Each string in the set of strings represented by the DFA's represents a sequence of events that a cell experiences from an initiating event (e.g. binding of a cytokine) to a final event (e.g. activation of a transcription factor). The initiating and final events are arbitrarily assigned. Consider each event in a string to possess some "purpose", somehow contributing to the overall event – the event of the initiating event leading to the final event. Each string would then represent "one way" in which a cell turns an initiating event to a final event. Each "way" could have some unique biological significance. For example, some events might represent cross talks to other pathways (i.e. these events initiate other pathways or impose some influences on other pathways). Therefore, a

unique combination of these events could represent a unique set of pathways being activated or influenced. A DFA, thus, represents a set of these "ways" a cell turn an initiating event to the final event.

One interesting criteria, for DFA alignment, would be to consider how each of these "ways" is represented or implemented in the other DFA. The underlying assumption is that each "way" used in one DFA is somehow used in the other DFA in some "form" – the idea used here is similar to the idea of aligning two amino acid sequences to infer functional similarity. The alignment of two DFA, then, consists of constructing a DFA that accepts all and only all of the following strings. For each string represented by the two DFA, take the best alignment with all the strings represented by the other DFA (i.e. not the alignment with the strings in the same DFA).

The alignment resulting from aligning two sequences of events (two strings) represents a "generic" sequence of events that captures the similarity between the two sequences of events. This is the rationale behind the pathway alignment algorithm used to align linear pathways as explained in chapter 3, "Part I: Abstraction to Linear Pathways". When applied to the alignment of DFA, the alignment, then, is a DFA that accepts all and only all "consensus" strings between all the strings represented by the two DFA being aligned. One could view this as a DFA that represents the "average behavior" between the two input DFA's.

One does not need to use all "consensus" strings (alignments) in the alignment DFA. If one uses only the set of "consensus" strings with alignment scores higher than some threshold value, then the alignment DFA would show the parts from both DFA's, which

are significantly similar (i.e. alignments higher than some threshold value) – or, it can be called the "consensus" DFA.

CAUTION: one always needs to keep the biological context in mind when analyzing the pathway DFA. In some cases, the same final state might represent different biological outcomes. This could occur when one is trying to align two DFA's that have final states representing different biological outcomes/conditions or when trying to minimize a DFA that has more than one final state representing different biological conditions. This is because the pathway alignment algorithm does not distinguish between the different biological conditions of each state.

The pathway (DFA) alignment algorithm used in this thesis is similar to the one used in PathBLAST [27] [28]. Please refer to section 2.1, "Conserved Pathways within Bacteria and Yeast".

4.2.1. Finding the Language of DFA without Cycles

The first step in the DFA alignment algorithm is to find all possible strings represented by the DFA's. Since the DFA's do not have any cycles, the number of strings they represent is always finite. For example, consider the following DFA.

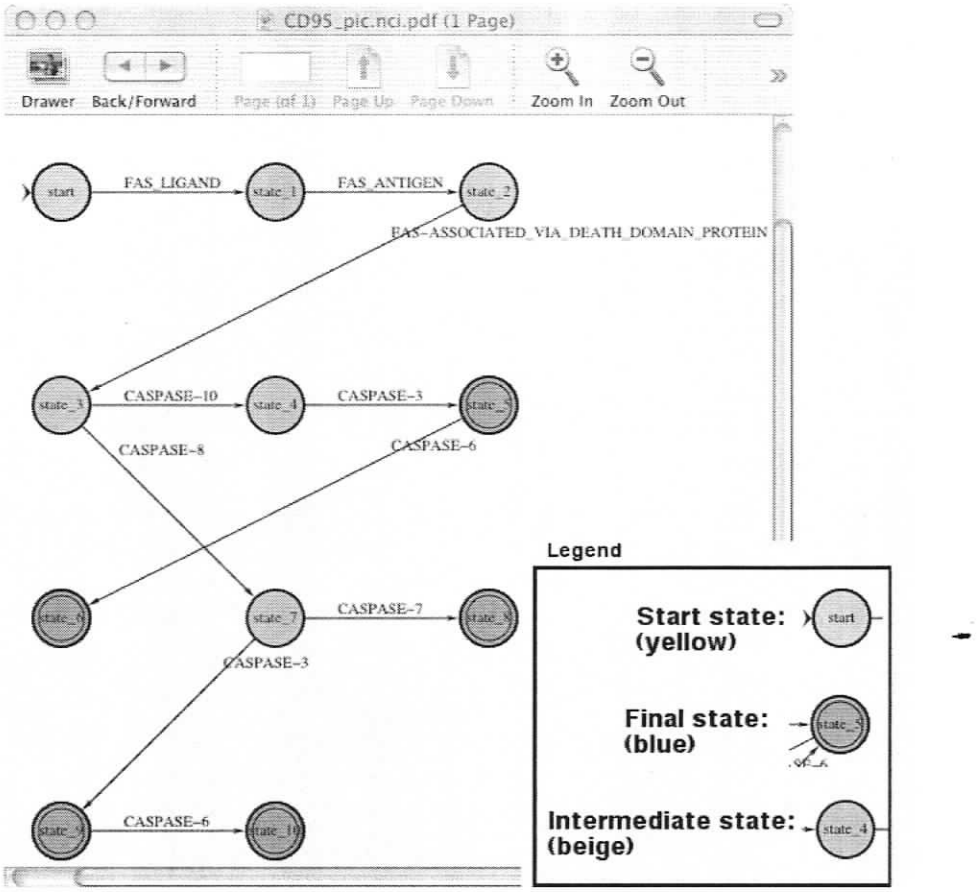


Figure 4-1: Fas-ligand signaling pathway represented by DFA.

The strings that the above DFA accepts are as follows:

FAS_L → FAS → FADD → Caspase-8 → Caspase-3
FAS_L → FAS → FADD → Caspase-8 → Caspase-3 → Caspase-6
FAS_L → FAS → FADD → Caspase-8 → Caspase-7
FAS_L → FAS → FADD → Caspase-10 → Caspase-3
FAS_L → FAS → FADD → Caspase-10 → Caspase-3 → Caspase-6

Table 4-1: Strings accepted by the DFA shown in Figure 4-1.

4.2.2. Pair-wise Linear Pathway Alignment

Let A be the set of strings accepted by pathway 1 and let a_i be the individual string in A.

Let B be the set of strings accepted by pathway 2 and let b_j be the individual string in B.

Let |A| denote the size of A and |B| denote the size of B. Let C be the set of pair-wise

alignments between all combination of strings in A and B, and let c_k be the individual alignment in C. The alignment is done using dynamic programming. The following is a pseudo code for constructing C.

```
int k=0;
for (int i=0; i<|A|; i++) {
  best_alignment
  for (int j=0; j<|B|; j++) {
    temp_alignment = pair-wise alignment between a_i and b_j.
    if temp_alignment score > best_alignment{
      best_alignment = temp_alignment
    }
  }
  let c_k = best_alignment;
  k++;
}
```

4.2.3. Constructing the "Consensus DFA"

The consensus DFA is a DFA that accepts all strings in C (the set of pair-wise alignments between all combination of strings in A and B) and only the strings in C.

4.2.3.1. Algorithm to Construct the "Consensus DFA"

The consensus DFA is constructed incrementally by successively adding every string (c_k) in C to the consensus DFA. The initial consensus DFA is an automaton that only accepts c_0. To do this, start with the start state, the start state can only accept the first element in c_0, which causes the state machine to transit to state 1, state 1 can only accept the second element in c_0, which causes the state machine to transit to state 2. The process continues until the last element in c_0 is used as the input symbol, which causes the state machine to transit to the final state. For all states in this initial consensus DFA, all "invalid" input symbols (i.e. symbols not found in c_0) will cause the state machine to transit to a dummy state. For all the other strings (c_k) in C, add c_k to the consensus DFA as follows. Starting

with the initial consensus DFA, find the longest substring (no gaps and the first element of this substring must be the first element in c_k) that the initial consensus DFA accepts. Let this substring be denoted as c_k.substring(0,l). In the state of the initial consensus DFA that accepts this substring, "branch off" and add new states to accept the rest of c_k (i.e. c_k.substring(l+1, $|c_k|$-1), where $|c_k|$ is the length of c_k). For all the newly added states, the state machine will transit to the dummy state for all input symbols not found in c_k.

When a "gap" occurs in c_k (an alignment between two strings), the state machine branches off. One branch accepts the generic protein input symbol; the other branch "waits" until the next "non-gap" input symbol occurs in c_k. If the end of c_k is reached, the state that is waiting for the next "non-gap" symbol becomes a final state. The following figure shows this algorithm.

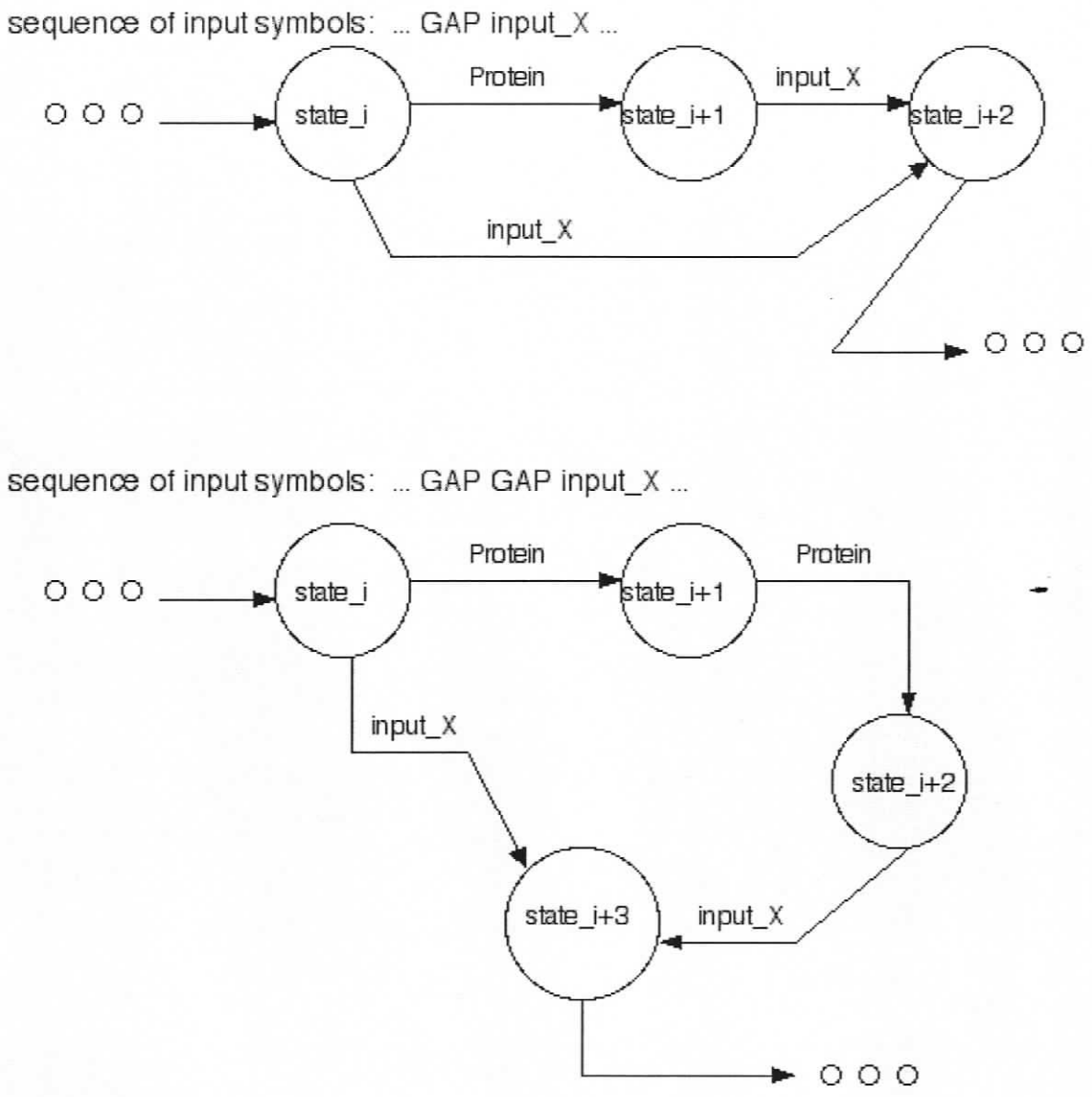


Figure 4-2: Algorithm to deal with gap when constructing a consensus DFA.

In the event of a sequence of consecutive "gap" input symbols in c_k, only the first state (i.e. the state before the first "gap" input symbol) would "wait" for the next "non-gap" input symbol – as shown in the lower part of Figure 4-2.

As mentioned before, the consensus DFA that accepts all strings in C represents a state machine that accepts all strings in A and B, where for an "aligned" transition, the input

symbol would be the lowest common parent between the input symbol in A and B. One does not need to use all strings in C. If one chooses a threshold, *T*, one can use a subset of strings in C such that the alignment score of c_k (C is a set of alignments) is greater than *T*. In this case, the consensus DFA would show a "common sub-state machine" between the two DFA being aligned.

4.2.3.2. Algorithm to Minimize the "Consensus DFA"

The consensus DFA constructed in the last section (4.2.3.1) might not be minimized. The algorithm described in [23] is used to minimize this DFA.

The following diagram summarizes the algorithm to align two DFA.

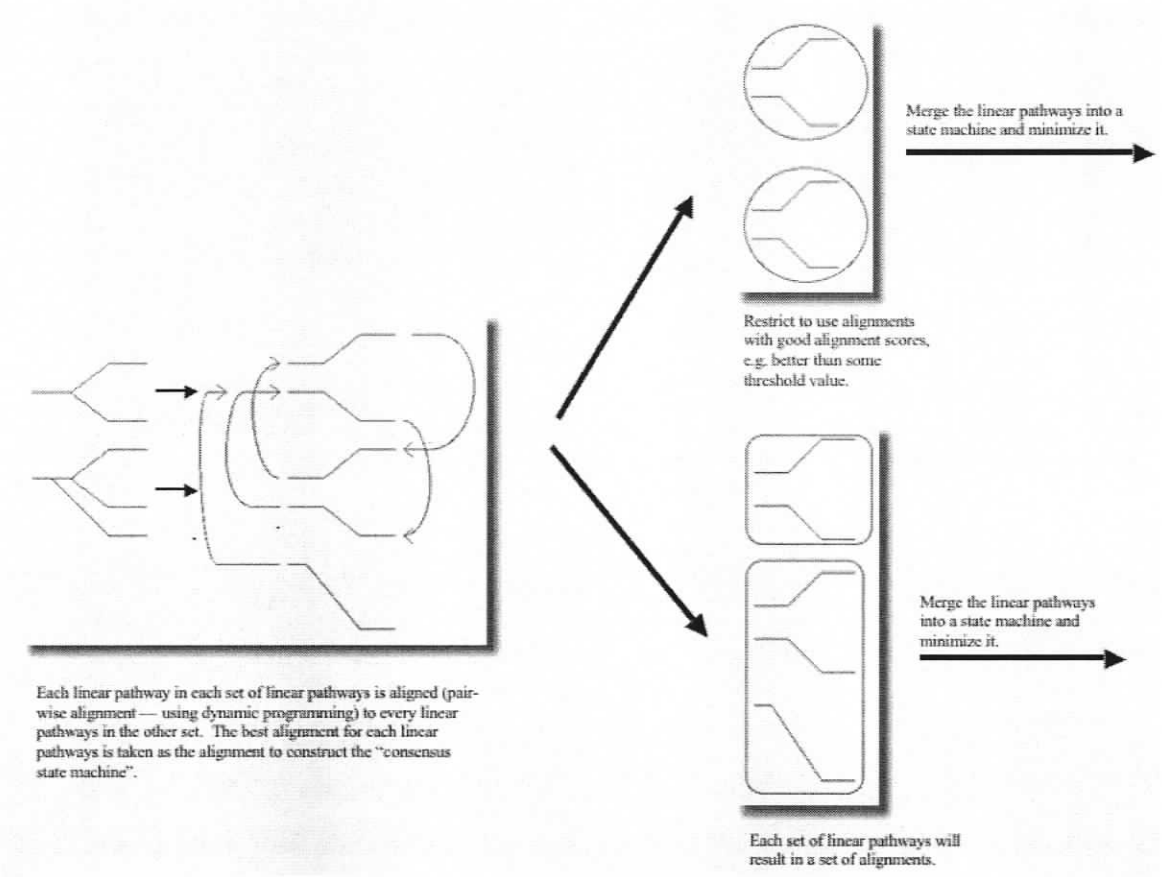


Figure 4-3: DFA alignment.

The following diagram shows how the input symbols are aligned in a consensus DFA.

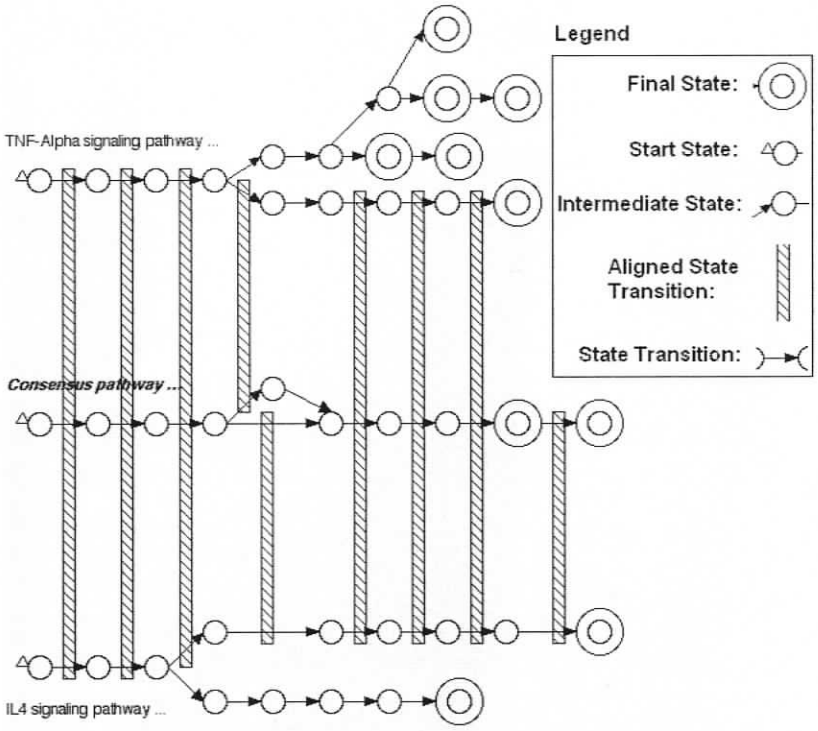


Figure 4-4: Input symbols are aligned in the consensus DFA.

The above consensus DFA only takes alignments with distance scores below some threshold – i.e. it shows the similar parts between the two DFA being aligned. The aligned input symbols are enclosed within the same striped-box. The double-circle represents final states while start states are represented by a circle with a triangle attached to it. Because of space constraints, the actual input symbols are not shown. Please refer to Figure 4-6, Figure 4-7 and Figure 4-17 for the corresponding DFA with input symbols shown. For TNFα signaling pathway DFA, please refer to Figure 4-6. For IL4 signaling pathway DFA, please refer to Figure 4-7. For the consensus DFA, please refer to Figure 4-17.

4.3. Sample Results

In this section, the signaling pathways examined in 3.7.1 will be analyzed using DFA's instead of strings of protein names.

4.3.1. Input Pathways

Please refer to 3.7.1 for the biology background of the following signaling pathways.

4.3.1.1. Apoptosis via Fas (CD95) ligand

The following diagram shows the Fas-ligand signaling pathway represented by a DFA without any "loops".

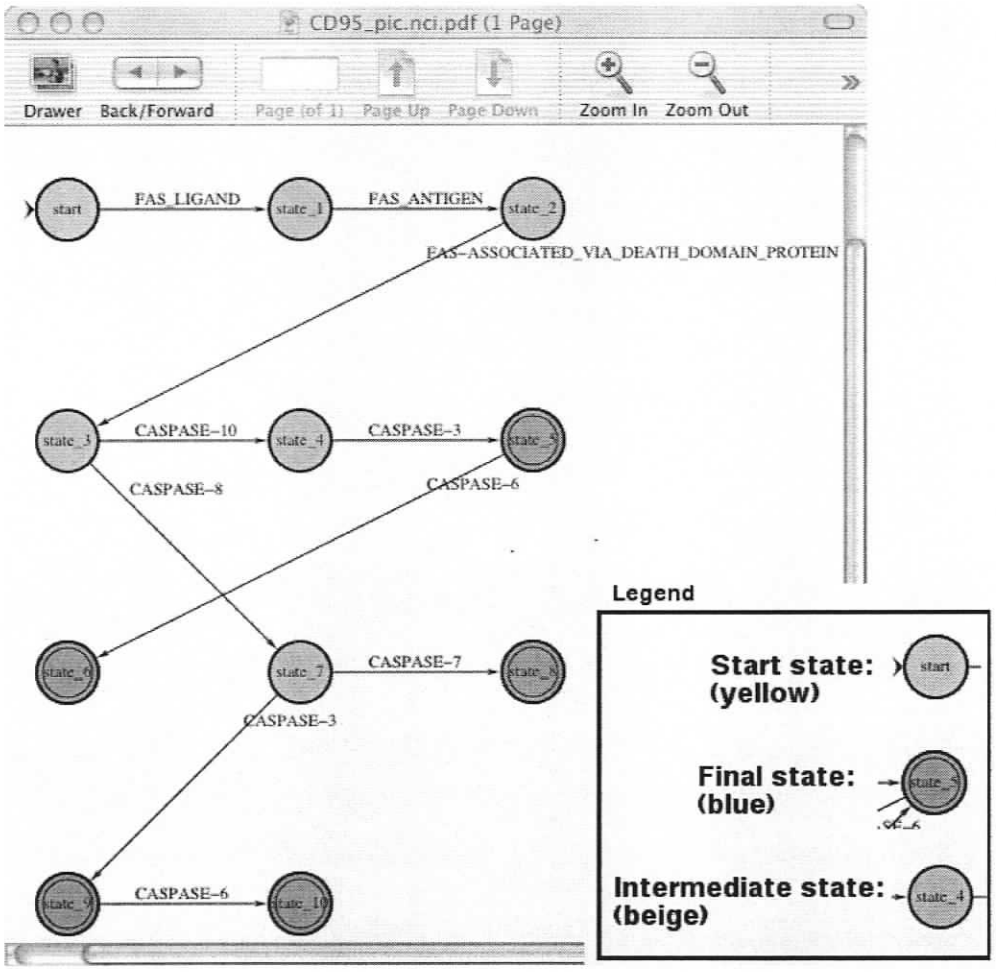


Figure 4-5: Fas-ligand signaling pathway.

4.3.1.2. Apoptosis via TNF-Alpha

The following diagram shows the TNFα signaling pathway represented by a DFA without any "loops".

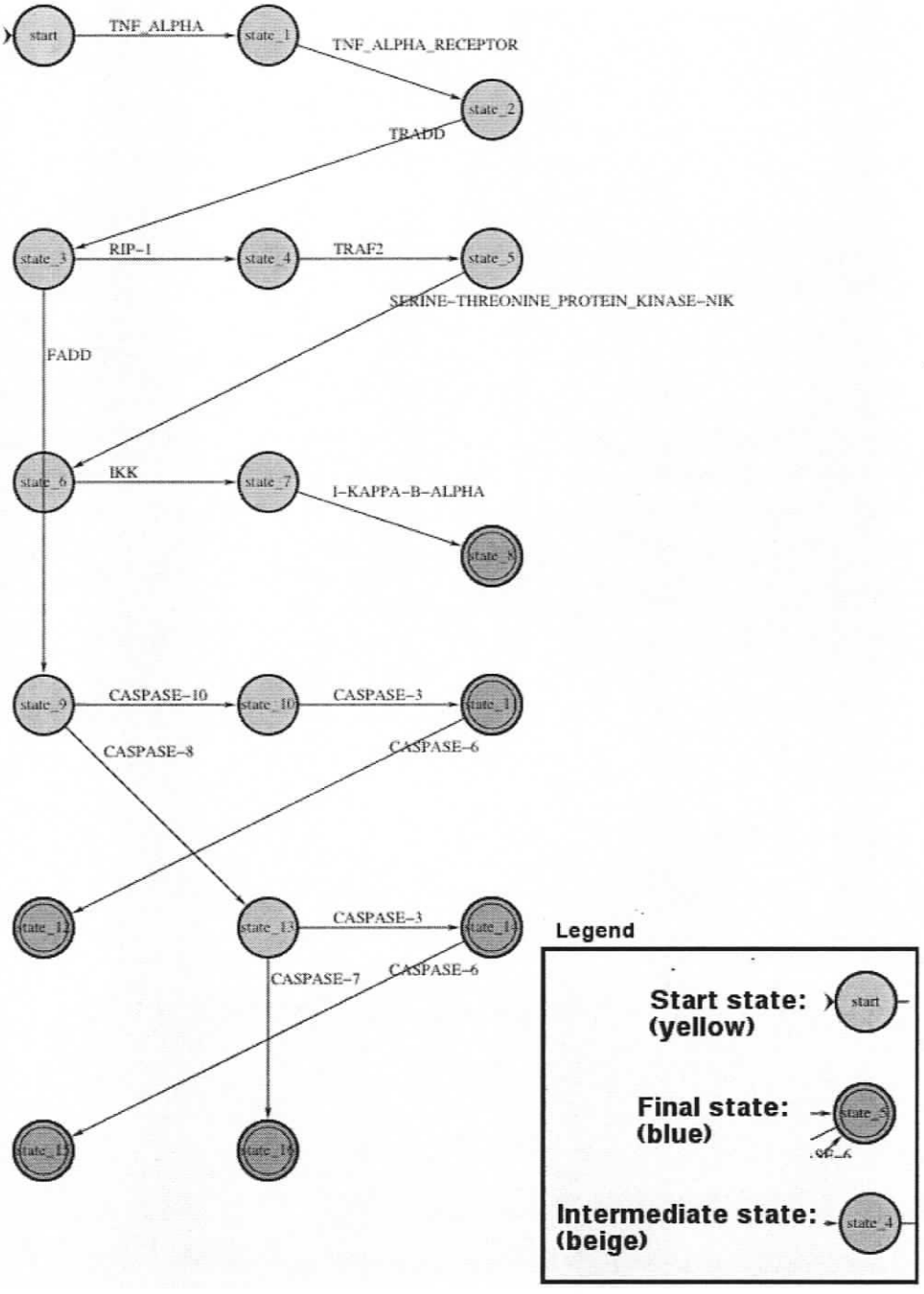


Figure 4-6: TNFα signaling pathway.

4.3.1.3. Interleukin-4

The following diagram shows the IL4 signaling pathway represented by a DFA without any "loops".

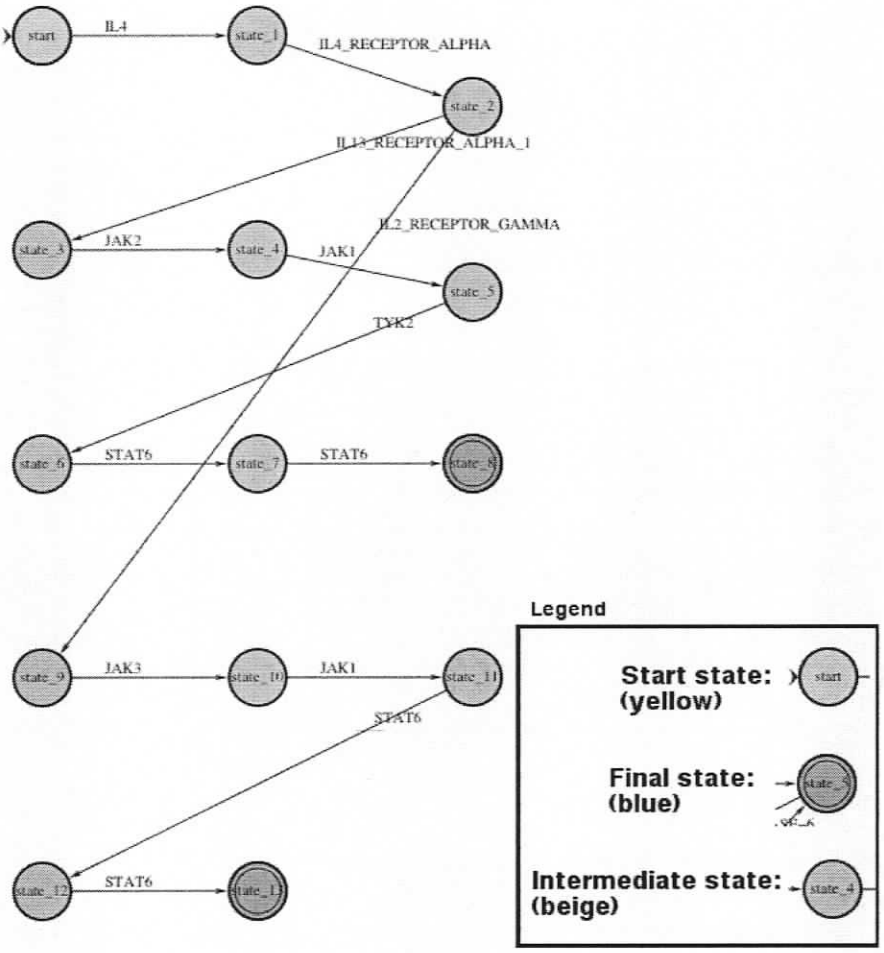


Figure 4-7: IL4 signaling pathway.

4.3.2. Pair-wise DFA Alignment

This section describes in detail the alignment process and the result.

4.3.2.1. Languages of Input DFA

The following lists the strings represented by the input pathways (DFA) – i.e. their languages.

Fas-ligand signaling pathway:

```
FAS_LIGAND FAS_ANTIGEN FADD CASPASE-10 CASPASE-3
FAS_LIGAND FAS_ANTIGEN FADD CASPASE-10 CASPASE-3 CASPASE-6
FAS_LIGAND FAS_ANTIGEN FADD CASPASE-8 CASPASE-3
FAS_LIGAND FAS_ANTIGEN FADD CASPASE-8 CASPASE-3 CASPASE-6
FAS_LIGAND FAS_ANTIGEN FADD CASPASE-8 CASPASE-7
```

TNFα signaling pathway:

```
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD RIP-1 TRAF2 NIK IKK I-KAPPA-B-ALPHA
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD FADD CASPASE-10 CASPASE-3
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD FADD CASPASE-10 CASPASE-3 CASPASE-6
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD FADD CASPASE-8 CASPASE-3
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD FADD CASPASE-8 CASPASE-3 CASPASE-6
TNF_ALPHA TNF_ALPHA_RECEPTOR TRADD FADD CASPASE-8 CASPASE-7
```

IL4 signaling pathway:

```
IL4 IL4_RECEPTOR_ALPHA IL13_RECEPTOR_ALPHA_1 JAK2 JAK1 TYK2 STAT6 STAT6
IL4 IL4_RECEPTOR_ALPHA IL2_RECEPTOR_GAMMA JAK3 JAK1 STAT6 STAT6
```

4.3.2.2. Pair-wise String Alignment

The following are the pair-wise string alignments of the strings in the Fas-ligand signaling pathway DFA and TNFα signaling pathway DFA. For display purposes, the NCI codes of the proteins in the alignments below are shown. Please refer to appendix A, "NCI Thesaurus Protein Code/Name Table" for the common names of the proteins. Please note, "------" represents a gap.

Fas-ligand signaling pathway:

```
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18287 C18031
Fas-ligand pathway: C20529 C17776 ------ C26106 C18287 C18031
 consensus pathway: C20500 C19285 ------ C26106 C18287 C18031

 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18287 C18031 C28436
Fas-ligand pathway: C20529 C17776 ------ C26106 C18287 C18031 C28436
 consensus pathway: C20500 C19285 ------ C26106 C18287 C18031 C28436

 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C28439
Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C28439
 consensus pathway: C20500 C19285 ------ C26106 C18182 C28439

 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C18031
Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C18031
```

```
 consensus pathway: C20500 C19285 ------ C26106 C18182 C18031

 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C18031 C28436
Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C18031 C28436
 consensus pathway: C20500 C19285 ------ C26106 C18182 C18031 C28436
```

TNFα signaling pathway:

```
Fas-ligand pathway: C20529 C17776 C26106 C18287 ------ C18031 ------ ------
 TNF_ALPHA pathway: C20535 C17800 C17907 C17923 C17812 C26487 MS0006 C17678
 consensus pathway: C20500 C19285 C26231 C16554 ------ C16554 ------ ------

Fas-ligand pathway: C20529 C17776 ------ C26106 C18287 C18031
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18287 C18031
 consensus pathway: C20500 C19285 ------ C26106 C18287 C18031

Fas-ligand pathway: C20529 C17776 ------ C26106 C18287 C18031 C28436
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18287 C18031 C28436
 consensus pathway: C20500 C19285 ------ C26106 C18287 C18031 C28436

Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C18031
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C18031
 consensus pathway: C20500 C19285 ------ C26106 C18182 C18031

Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C18031 C28436
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C18031 C28436
 consensus pathway: C20500 C19285 ------ C26106 C18182 C18031 C28436

Fas-ligand pathway: C20529 C17776 ------ C26106 C18182 C28439
 TNF_ALPHA pathway: C20535 C17800 C17907 C26106 C18182 C28439
 consensus pathway: C20500 C19285 ------ C26106 C18182 C28439
```

4.3.2.3. Construct Consensus DFA

This section shows the consensus DFA's. The following DFA alignments were done:

- Fas-ligand signaling pathway against TNFα signaling pathway
- Fas-ligand signaling pathway against IL4 signaling pathway
- TNFα signaling pathway against IL4 signaling pathway

First, a consensus DFA that takes in all alignments is constructed. Next, a consensus DFA that takes in alignments above some threshold alignment score is constructed. Please note, DFA alignment is a symmetric operation – i.e. aligning DFA_1 against DFA_2 generates the same consensus DFA as aligning DFA_2 against DFA_1.

4.3.2.3.1. Consensus DFA: Fas-ligand against TNFα signaling pathway

The following is the Fas-ligand signaling pathway against TNFα signaling pathway consensus DFA taking all alignments. Please note, the input symbol "------" indicates a non-aligned input symbol.

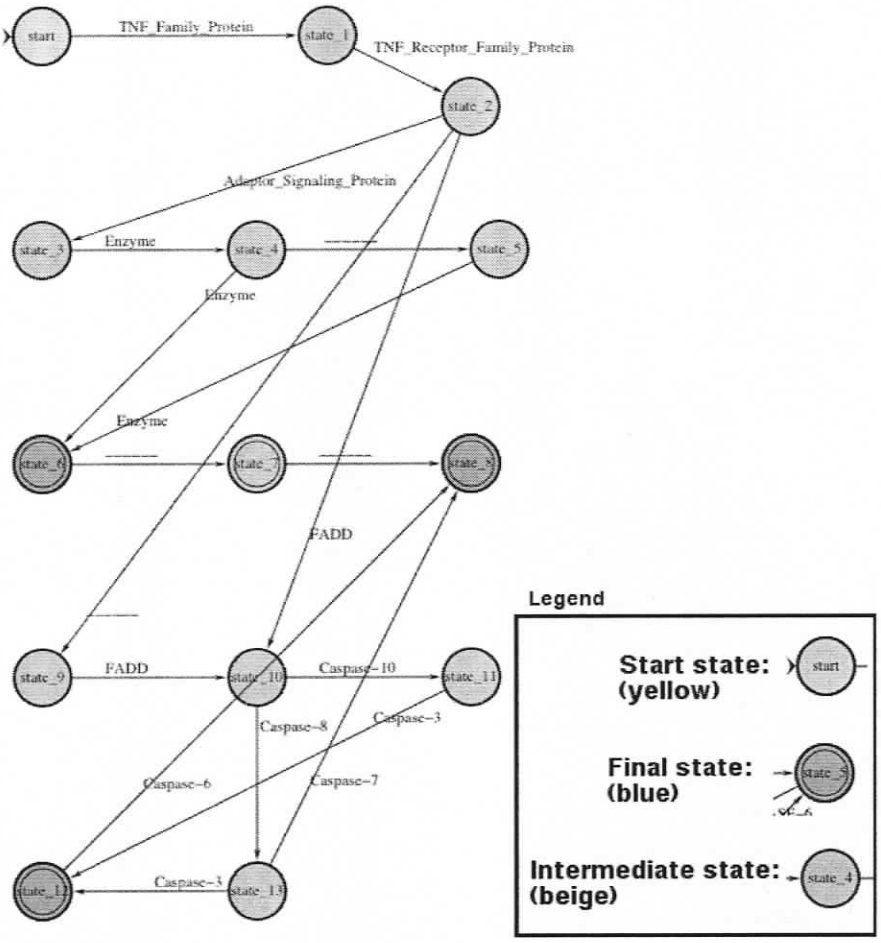


Figure 4-8: Fas-ligand DFA aligned against TNFα DFA, taking all alignments.

The following is the Fas-ligand signaling pathway against TNFα signaling pathway consensus DFA taking alignments with *distance* score *below* a threshold, which is set so that about 50% of the alignments were cut off.

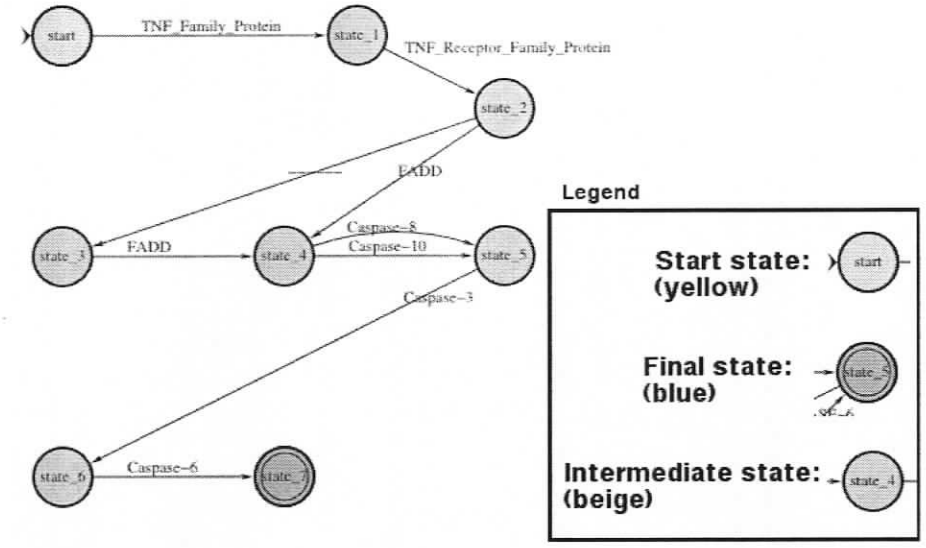


Figure 4-9: Fas-ligand DFA aligned against TNFα DFA, taking "good" alignments.

Significant parts of the Fas-ligand and TNFα pathways are very similar (in fact, they are the same). The similar parts involve the activation of the different caspases. Both pathways activate the same caspases. The pathways differ more "before" the initiator caspases (Caspase8, Caspase10) are activated. This could suggest that these pathways are controlled differently in terms of how and when the binding of the cytokine (Fas-ligand and TNFα) cause the activation of the initiator caspases. However, once the initiator caspases are activated, both pathways would produce the same effect.

The following two figures show parts of the Fas-ligand and TNFα DFA that took part in the above consensus DFA (Figure 4-9). In other words, these are parts of the DFA that are significantly similar to each other.

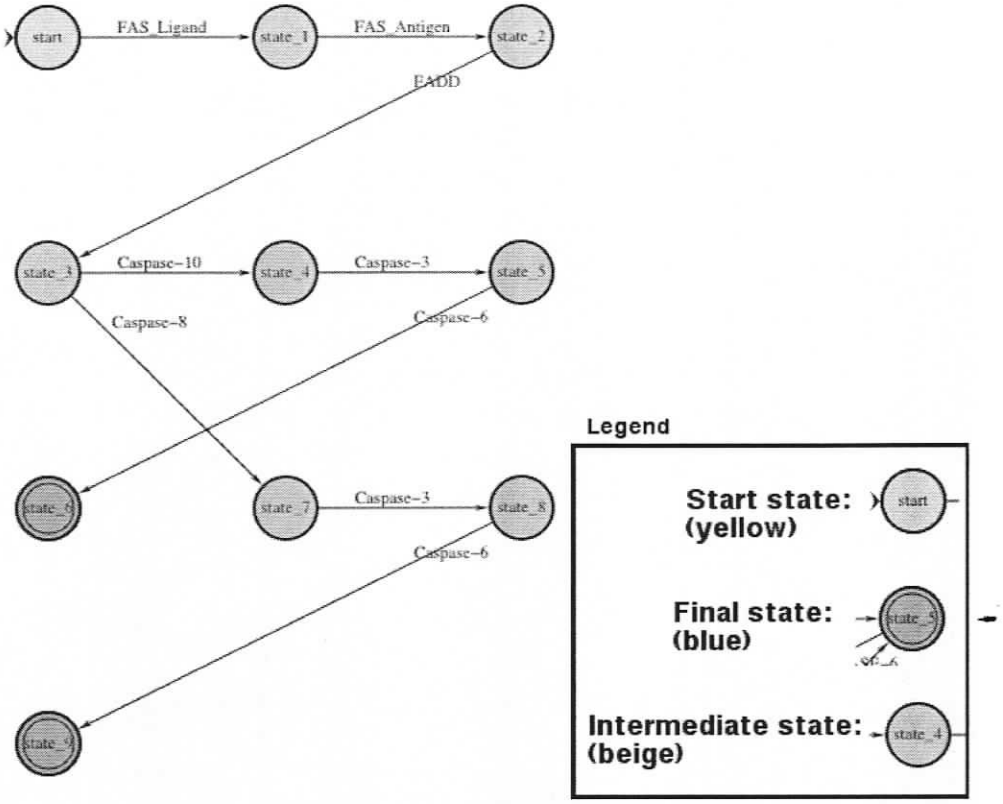


Figure 4-10: Fas-ligand pathway significantly similar to TNFα pathway.

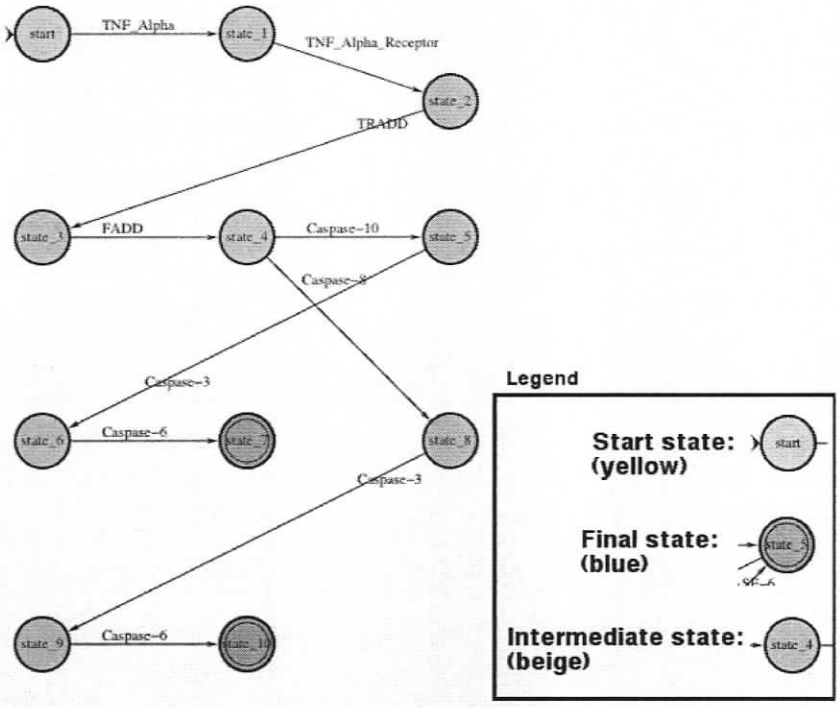


Figure 4-11: TNFα pathway significantly similar to Fas-ligand pathway.

4.3.2.3.2. Consensus DFA: Fas-ligand against IL4 signaling pathway

The following is the Fas-ligand signaling pathway against IL4 signaling pathway consensus DFA taking all alignments.

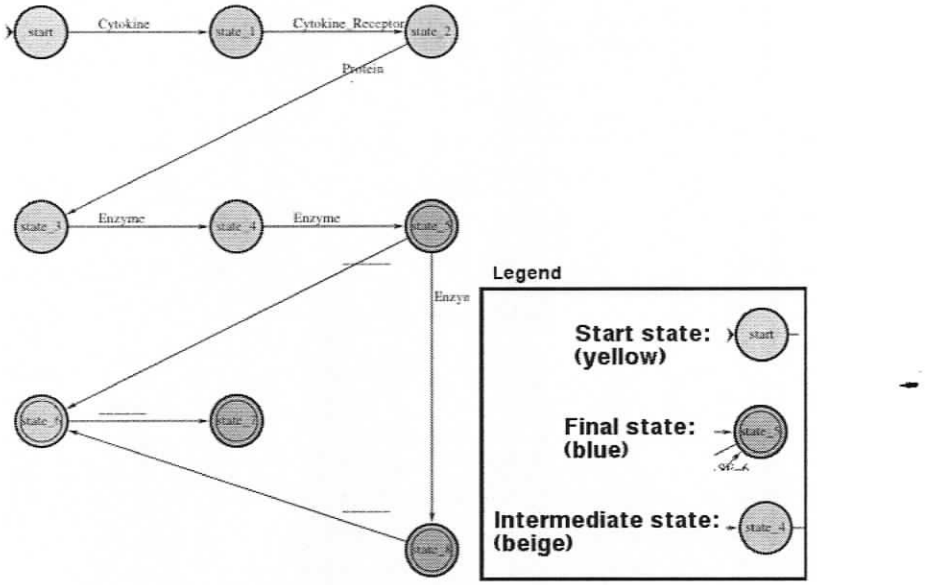


Figure 4-12: Fas-ligand DFA aligned against IL4 DFA, taking all alignments.

The following is the Fas-ligand signaling pathway against IL4 signaling pathway consensus DFA taking alignments with *distance* score *below* a threshold. The threshold is set so that about 50% of the alignments were cut off.

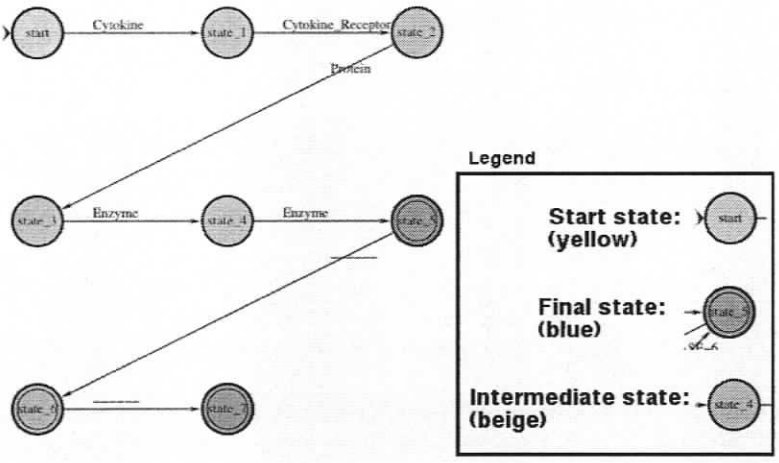


Figure 4-13: Fas-ligand DFA aligned against IL4 DFA, taking "good" alignments.

Unlike the alignment between Fas-ligand and TNFα pathways, which is an alignment between pathways with very similar biological function, the alignment between Fas-lignad and IL4 pathways involves pathways that have different biological functions. The following two figures show parts of the Fas-ligand and IL4 DFA that took part in the above consensus DFA (Figure 4-13). In other words, these are parts of the DFA that are significantly similar to each other.

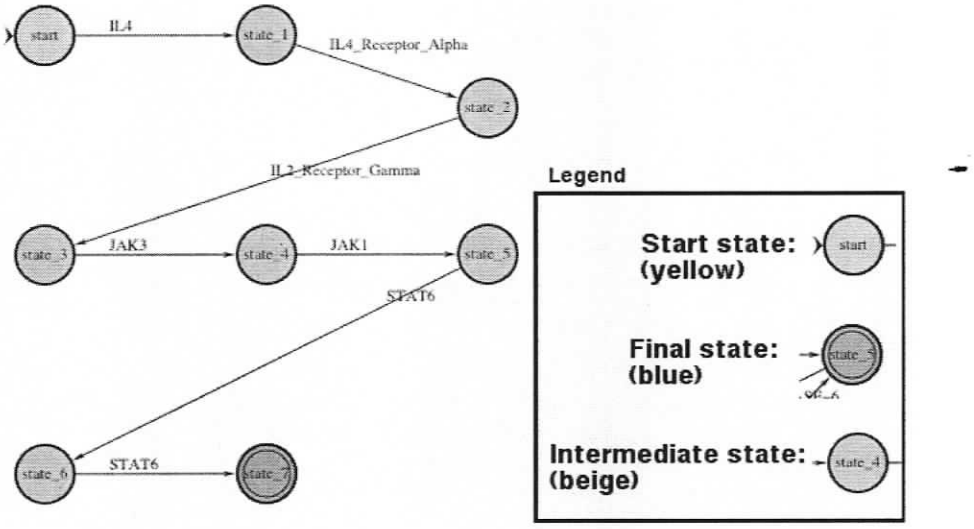


Figure 4-14: IL4 pathway significantly similar to Fas-ligand pathway.

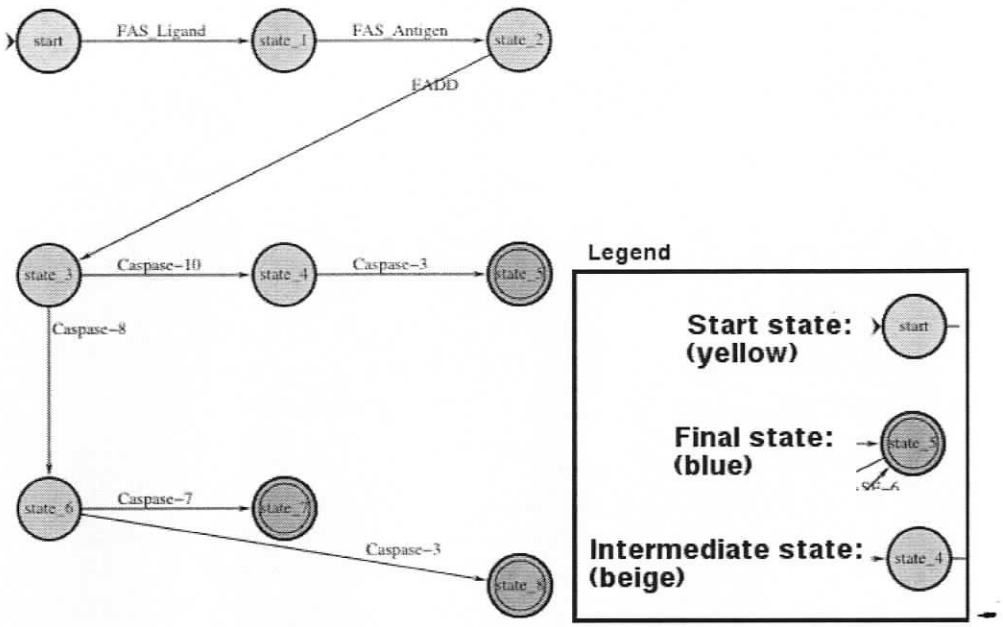


Figure 4-15: Fas-ligand pathway significantly similar to IL4 pathway.

Even though the two pathways have quite different biological function (Fas-ligand is associated with apoptosis, while IL4 pathway is associated with immune response – activation of transcription), the following "pathway motif" is common between them.

`Cytokine → Cytokine_Receptor → Protein → Enzyme → Enzyme`

The interesting motif is "`Enzyme → Enzyme`". Looking at the linear pathway alignments (done as part of the DFA alignment algorithm), one could observe how this "pathway motif" came about.

```
       IL4 pathway: JAK3       JAK1
Fas-ligand pathway: Caspase-10 Caspase-3
Fas-ligand pathway: Caspase-8  Caspase-7
Fas-ligand pathway: Caspase-8  Caspase-3
 Consensus pathway: Enzyme     Enzyme
```

One has to keep in mind the following. Since the DFA model treats concurrent events as arbitrary interleaving of sequential events, one has to keep in mind that while JAK3 and JAK1 are activated concurrently, the initiator caspases (Caspase-10 and Caspase-8)

activate the effector caspases (Caspase-3 and Caspase-7). However, the following biological similarity among the pathway segments that contribute to this motif can be suggested. Caspases activate downstream molecular machinery to carry out apoptosis, and JAK's activate STAT's – transcription factors to carry out transcription. Thus, in this sense, caspases and JAK's do have similar biological function, and the pathway alignment algorithm was able to select this "pathway motif".

4.3.2.3.3. Consensus DFA: TNFα against IL4 signaling pathway

The following is the TNFα signaling pathway against IL4 signaling pathway consensus DFA taking all alignments.

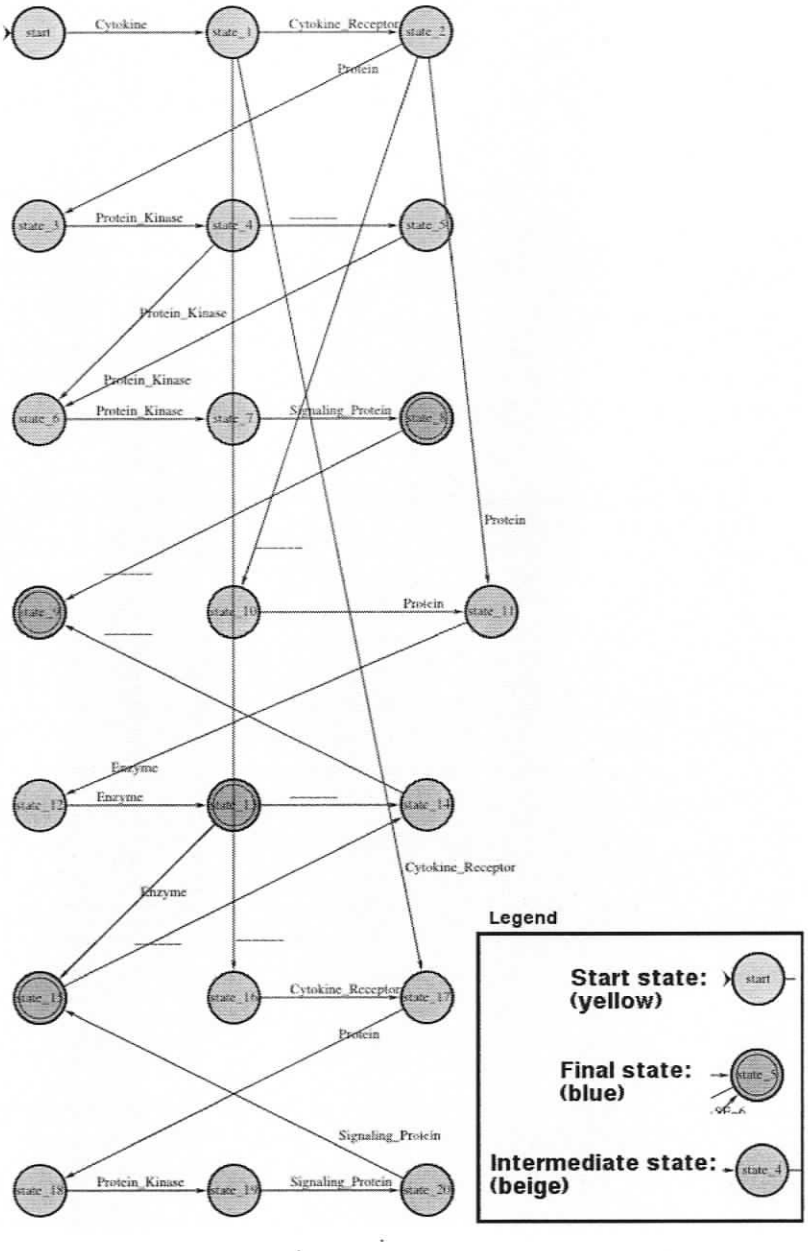


Figure 4-16: TNFα DFA aligned against IL4 DFA, taking all alignments.

The following is the TNFα signaling pathway against IL4 signaling pathway consensus DFA taking alignments with *distance* score *below* a threshold. The threshold is set so that about 75% of the alignments were cut off.

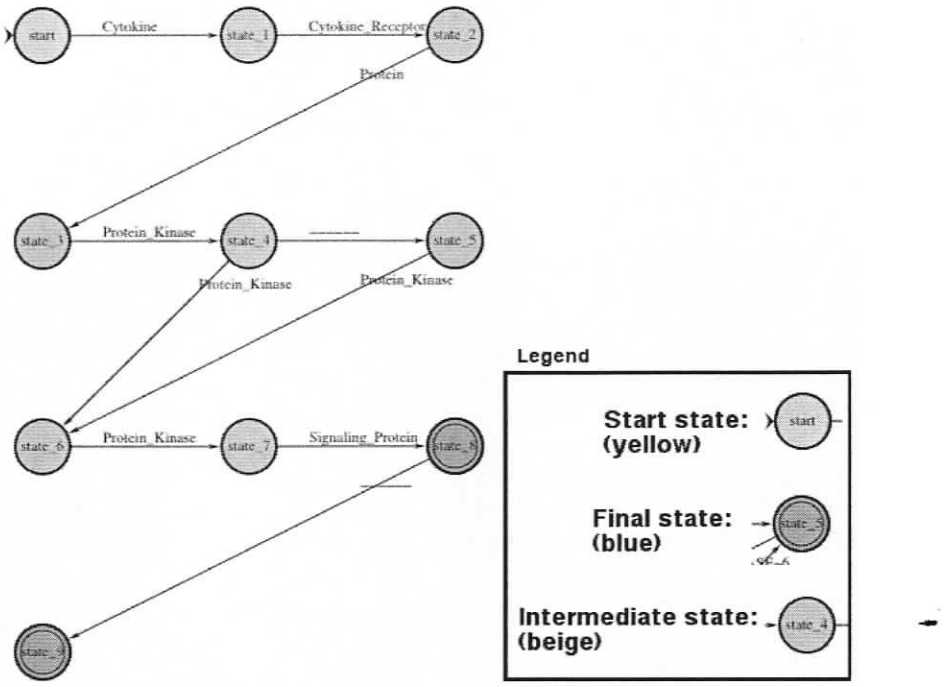


Figure 4-17: TNFα DFA aligned against IL4 DFA, taking "good" alignments.

The following two figures show parts of the IL4 and TNFα DFA that took part in the above consensus DFA (Figure 4-17). In other words, these are parts of the DFA that are significantly similar to each other.

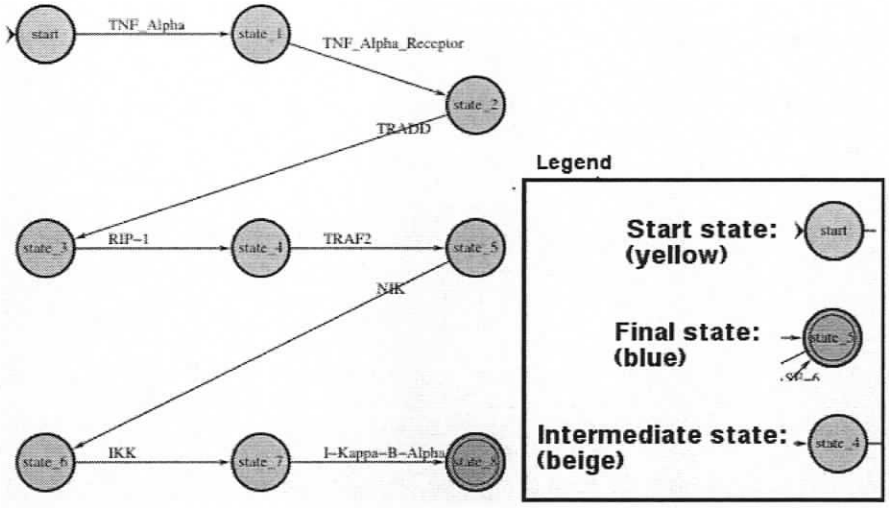


Figure 4-18: TNFα pathway significantly similar to IL4 pathway.

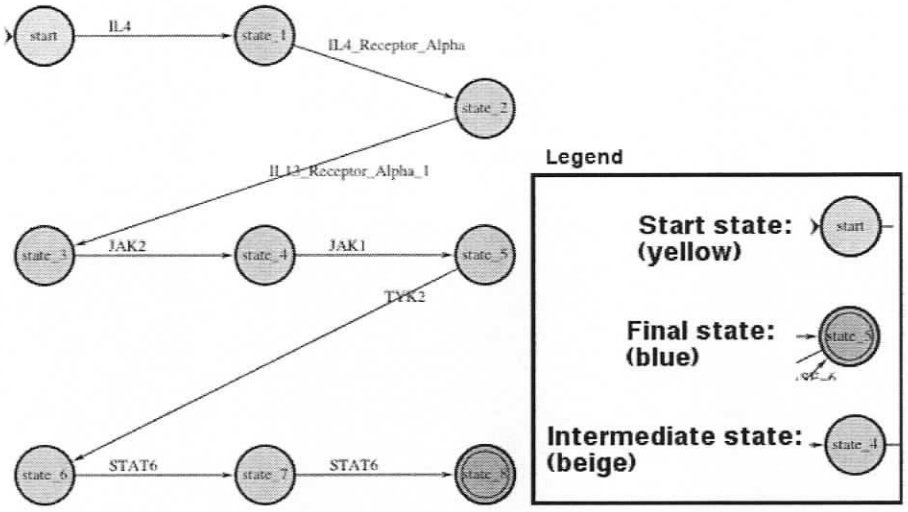


Figure 4-19: IL4 pathway significantly similar to TNFα pathway.

One interesting observation in the above three diagrams (Figure 4-17, Figure 4-18, and Figure 4-19) is that the parts of the input pathways (Figure 4-18 and Figure 4-19) contributing to the consensus DFA both deal with initiation of transcription. In other words, the DFA alignment algorithm is able to extract parts of the pathways with similar biological function. In the TNFα pathway, one "branch" of the pathway leads to apoptosis, while the other "branch" of the pathway leads to transcription. Please refer to section 3.7.1.3, "Apoptosis Signaled by TNF-Alpha". Looking at the linear pathway alignments (done as part of the DFA alignment algorithm), one could observe the following interesting "pathway motif" occurring between the two transcription-activating signaling pathway:

```
 TNFα pathway: NIK             IKK             I-Kappa-B-Alpha
  IL4 pathway: JAK1            TYK2            STAT6
Pathway motif: Protein_Kinase Protein_Kinase Signaling_Protein
```

It would be very interesting if more transcription-activating pathways have the above pathway motif. If so, there could be some biological significance of having such a motif.

4.4. Mini-Conclusion

The contributions made in the research described in chapter 4 are as follows.

- A pathway representation model was developed using deterministic finite automata.
- An algorithm was developed to align two pathways represented as DFA.
- Three biochemical pathways were represented as DFA and were aligned with each other using the pair wise alignment method developed in this chapter. Interesting pathway motifs were observed in the pathway alignment between Fas-ligand signaling pathway and IL4 signaling pathway, and the alignment between TNFα signaling pathway and IL4 signaling pathway.

Chapter 5

5. Part III: Limitation (Evaluation) of the Model and Future Works

This chapter describes some limitations of the biochemical pathway representation model developed in this thesis. It also presents some ideas in dealing with some of these limitations as future works.

As mentioned earlier in this thesis (section 3.1), the biochemical pathway representation model used in this thesis is very simplistic. It ignores many aspects of the different biochemical pathways. For example, the thermodynamics and the kinetics of the pathways are ignored – except for the stoichiometry. Therefore, when one is working with this model, one has to keep in mind that the model represents a set of pathways that are theoretically possible based on stoichiometry only. It is possible that some or all of these pathways do not obey the laws of thermodynamics or kinetically very slow such that practically these chemical reactions would never occur. Given the above limitations of the model, one must be very cautious when generating pathway models based on pathway information from the literature. One must take special care that models generated are consistent with existing literature.

The following list describes some specific limitations of the pathway representation model. It also provides some future research ideas.

1. DFA's without loops does not deal with feedback mechanisms, which are very common in biological systems. The reason why DFA's with loops are difficult to deal with in this thesis is that the size of the language (i.e. the number of strings) that a DFA with loops represents is infinite. Hence, the DFA alignment algorithm

developed in this thesis would not work in such cases, since the algorithm tries to do pair wise alignment with all the strings represented by the input DFA's.

One solution to this problem would involve associating a probability function with every input symbols. Consider the following state machine.

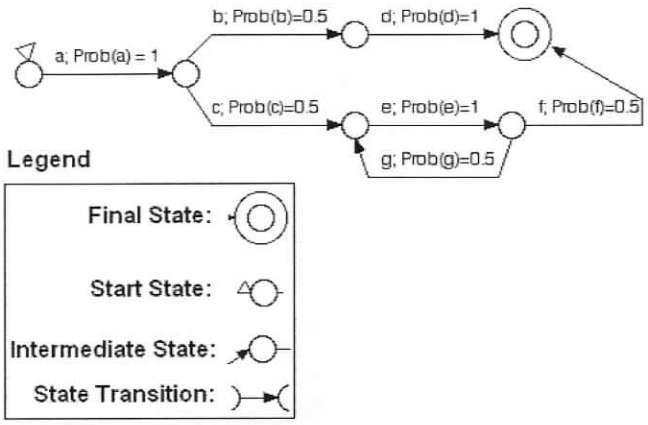


Figure 5-1: DFA with probability function associated with input symbol.

For each state, the probabilities of accepting any valid input symbols can be the same[18] and the sum of the probabilities of accepting all valid input symbols is always one[19]. One could then associate a probability with every string represented by this automaton. For example …

Prob("abd") = 0.5
Prob("acef") = 0.25
Prob("acegef") = 0.125
… etc

The sum of the probability of all strings must equal one. This is shown by the following calculation. There are two sets of strings represented by the automaton shown in Figure 5-1.

---

[18] The probabilities of accepting different valid input symbols can be different if it is supported by biological evidence – e.g. in state *x*, most of the time reaction with protein *a* would be observed and very rarely, reaction with protein *b* would occur.

[19] The probability of accepting an invalid input symbol, which would transit to the dummy state, is always zero

"abd"
"ac(eg)^{n}ef" for $0 \le n \le \infty$

The probabilities of the above strings are as follows:

$$\text{Prob("abd")} = 0.5$$

$$\text{Prob("ac(eg)}^n\text{ef")} = 0.5 \cdot 0.5 \cdot \sum_{i=0}^{\infty} 0.5^i = \frac{0.5 \cdot 0.5}{1 - 0.5} = 0.5$$

For the DFA alignment algorithm, one could set a threshold value, t, such that all strings with probability less than t would be ignored. Now, the number of strings to compare becomes finite. This method would prevent "repeats" (due to loops in the automaton) to be over-represented in the language. For example, "ac(eg)^{n}ef" represents an infinite number of strings. During pair wise alignment, the strings should also be weighted according to their probabilities as described above – i.e. an alignment between two strings, both have high probability, should have a higher score (more weight) than an alignment between two strings either of which has a lower probability. Further research will be needed to show whether this method can produce biologically significant results.

2. Concurrent events are treated as arbitrary interleaving of sequential events. This is obviously not a realistic view of a biological system. One would need to extend the model to deal with concurrent events. One solution would be to consider all possible combination of interleaving of sequential events. For example, consider the three input symbols, a, b, and c. If these input symbols represent concurrent events, the following six automata would need to be used.

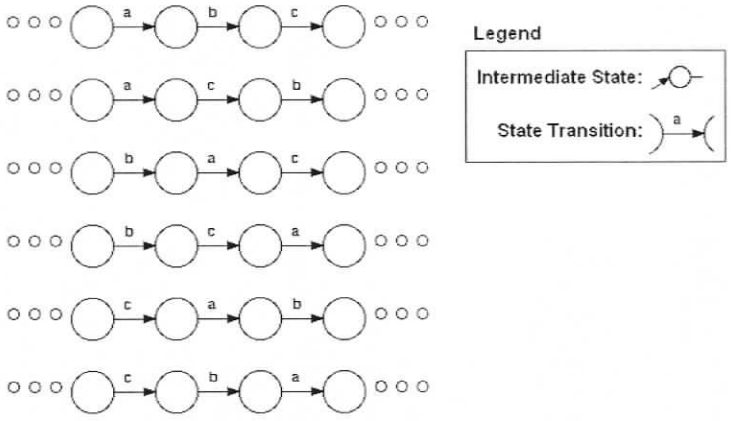


Figure 5-2: Automata for concurrent events a, b, and c.

One problem with this solution would be scalability. The number of automata needed would explode even for small number of concurrent events. For n concurrent events, one would need $n!$ automata.

3. The pathway representation model does not capture how molecules actually "work together". Molecules can "work together" in many different ways. The following presents some examples.

    - One molecule activates another. For example, a tyrosine kinase transfers a phosphate group to a tyrosine residue in another protein. Molecules can activate other molecules in other ways as well. For example, they could bind, form covalent bonds, cause confirmation changes … etc.
    - Molecules coactivate each other. For example, it is currently believed [21] that caspase-8 is activated via "induced proximity". The "induced proximity" model suggests the following. The adaptor molecules (e.g. FADD), which are bound to membrane-bound signaling complexes, recruit procaspase-8 (an inactive form of caspase-8). This would result in a high local concentration of

procaspase-8 molecules, which causes procaspase-8 molecules to coactivate each other.

To deal with this problem, one would need to construct an ontology of molecular interactions. Pathway comparison would then involve comparison between how the interactions between molecules differ from each other in addition to how molecules differ from each other.

4. Start and final states are arbitrarily determined. One must be cautious when analyzing these pathway automata, as their start and final states might refer to completely different biological conditions. Hence, the conclusions drawn from them might be biologically meaningless.
5. The model does not allow specification of concentrations or amounts of chemicals (input symbols) introduced. It could be possible that the biological system will react differently when different amounts of chemicals are introduced. For example, consider two different chemicals (A and B) that have one common biological function – e.g. able to bind to a certain region of DNA to promote transcription. Suppose the concentration of either A or B needed to promote transcription is x. If only A or B, but not both, is present and if the concentration of A is $0.5x$ and the concentration of B is $0.5x$, transcription would not be promoted. However, if both A and B are present and their concentration are both $0.5x$, then transcription would be promoted.

Chapter 6

6. Part IV: Pathway Comparison Software Suit

This chapter describes the software tools and their design.

6.1. System Overview (Architecture)

This subsection describes the software system architecture.

6.1.1. Main Components

The software system is composed of two components: a MySQL [60] database and a Java [49] application. The following diagram depicts the software architecture.

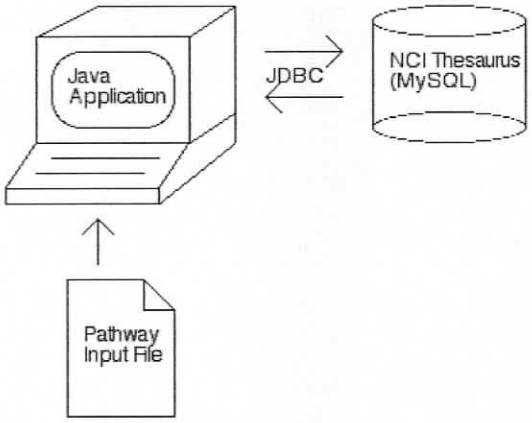


Figure 6-1: Software Architecture.

As shown in the above diagram, the Java application reads the input pathways from an input text file and compares the pathways using the NCI Thesaurus stored in a local MySQL database. The NCI Thesaurus is imported to the local MySQL database via the cancer Bioinformatics Infrastructure Objects (caBIO) library [6] provided by NCI. The

Java application interacts with the MySQL database via the JDBC API (Java Database Connectivity Application Program Interface) [50].

6.1.2. Class Overview

The software system consists of the following Java class packages.

- `pathwayAlign.mysqlTables` – this package contains the utility for extracting information from the NCI Thesaurus and stores it in the local MySQL tables.
- `pathwayAlign.matchNode` – this package contains the utility for mapping common protein names to specific protein codes/names in the NCI Thesaurus. This utility is used to preprocess the input files. Please refer to "D.3.2 Preprocess Input Files" on how to preprocess the input files.
- `pathwayAlign.align` – this package contains the classes responsible for comparing pathways represented by a sequence of strings.
- `pathwayAlign.automata` – this package contains the classes responsible for comparing pathways represented by DFA's.

Please refer to "Appendix C: Pathway Comparison Software Suite, Directory Structure" for a detail description of the packages.

6.2. Similarity Matrix

This subsection describes the software design of the feature that provides the similarity matrix function. The entries of the similarity matrix are calculated "on the fly" from a local version of the NCI Thesaurus as described in section "3.3.2.3 Similarity Score Calculation". The NCI Thesaurus is stored in a local MySQL database. One might

wonder why one would store the NCI Thesaurus, a tree structure, in a relational database. It would seem more natural to store the NCI Thesaurus in an xml file, which is naturally a tree structure. In fact, one could download the NCI Thesaurus as an xml file. The reason for using a MySQL database is scalability. If one were to use an xml file for storing the NCI Thesaurus, one would have to load the entire NCI Thesaurus into the Java application during the pathway analysis. This would impose an increasing amount of memory requirement on the Java application as the NCI Thesaurus grows – i.e. classifies more concepts (e.g. proteins).

6.2.1. MySQL Tables

The following figure shows the MySQL tables.

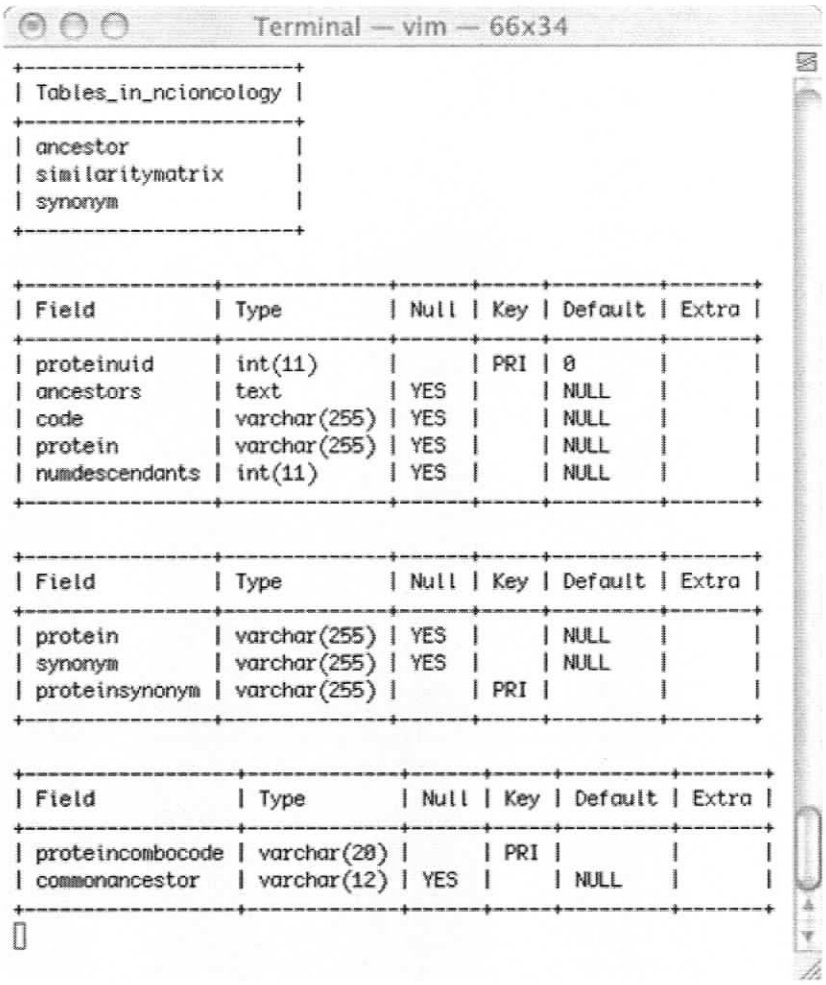


Figure 6-2: MySQL Tables.

Table: ancestor

The table `ancestor` is used to store the NCI Thesaurus tree structure. The field, `proteinuid`, is the primary key. Its only purpose is to maintain the integrity of the table, otherwise it is not used. The field `code` stores the unique identifier (UID) used in the NCI Thesaurus to index the different concepts. For example, the UID of "Caspase-3" is "C18031". The field, `ancestors`, stores a string representing the ancestry line of possibly one of its immediate superconcepts. For example, the ancestry line of "Caspase-3" is as follows. "Capase-3" is a "Caspase", which is a "Cysteine Proteinase", which is a

"Protease", which is a "Hydrolase", which is an "Enzyme", which is a "Protein". Thus, the `ancestors` field is:

`C20027;C16554;C16701;C16965;C16481;C18153.`

It is possible that a concept have more than one immediate ancestor. In this case, there would be *n* rows in the `ancestor` table for this particular concept, where *n* is the number of immediate ancestors. For example, a protein, "signal transducer and activator of transcription" (STAT), is classified both as a "signaling protein" and as a "transcription factor". Therefore, there are two rows for STAT in the `ancestor` table. The field `protein` is used to store the "official" protein name used in the NCI Thesaurus. The field `numDescendants` is used to store the number of subconcepts, under this concept, that have no subconcepts. If one regards the NCI Thesaurus as a tree, the `numDescendants` field represents the number of leaf nodes under this node.

***Table:* synonym**

The table `synonym` is used to store the synonym of the different concepts. For example, in the NCI Thesaurus, "Caspase-3" has the following synonyms:

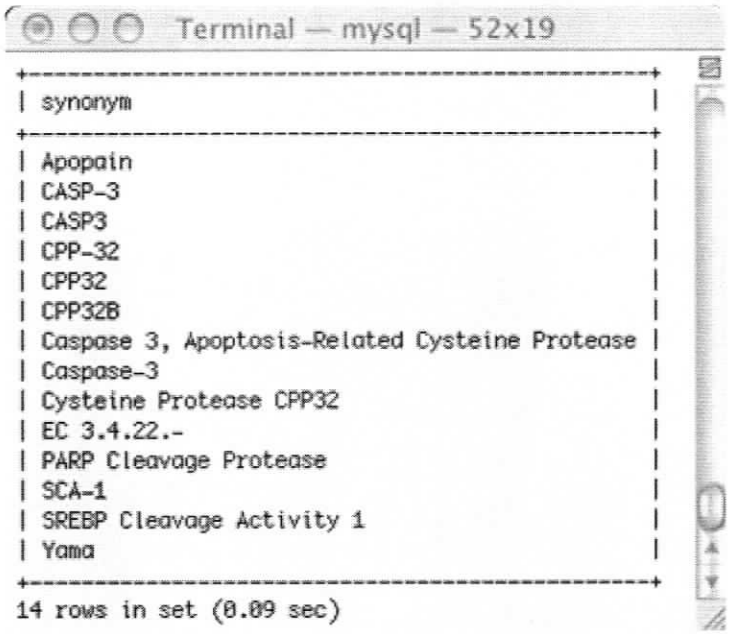


Figure 6-3: Synonyms of "Caspase-3".

Therefore, "Caspase-3" has 14 rows in the `synonym` table. The field `synonym` stores the synonym and the field `protein` stores the "official" protein name used in the NCI Thesaurus. The field `proteinsynonym` functions as a primary key and it is used only to maintain the table's integrity. This field is not used in any other way.

***Table:* `similaritymatrix`**

The table `similaritymatrix` is used to store temporary results used during the pathway comparison analysis. One expensive operation that is used very often during pathway comparison is: given two proteins, find out their common parent/ancestor in the NCI Thesaurus. Whenever one such operation is performed, the results are stored in the `similaritymatrix` table so that in the future, when this operation is performed on the same proteins, no re-calculations are needed. The field, `commonancestor`, stores the NCI Thesaurus UID of the common parent/ancestor of the two proteins, and the field,

proteincombocode, is a concatenation of the NCI Thesaurus UID of that two proteins.

6.2.2. Import from NCI Thesaurus

The NCI Thesaurus was imported to the local MySQL database via the cancer Bioinformatics Infrastructure Objects (caBIO) library [6]. This library uses Java's remote method invocation (rmi). For the pathway comparison analysis done in this thesis, only part of the NCI Thesaurus is needed – "Protein, Organized by Function". Thus, only this part of the NCI Thesaurus was imported to the local MySQL database. The fields in the ancestor table – ancestors and numDescendants – are calculated while traversing through the NCI Thesaurus during this import process.

6.2.3. Lowest Common Parent

Finding the lowest common parent is one of the most common operations during pathway comparison analysis. It is done by comparing the ancestors field of the ancestor table of the two input proteins, and returning the matched protein code furthest to the right. For example, given the two proteins and their respective ancestors field value:

```
Caspase-3: C20027;C16554;C16701;C16965;C16481;C18153
Fatty_Acid_Synthase: C20027;C16554;C17210;C16259
```

The lowest common parent has protein code: C16554, which represents "Enzyme". If one starts traversing the ancestors field value from the right (i.e. from the protein code of the immediate superconcept), one would be able to find the lowest common parent in O(n) time where n is the length of the longest ancestors field value.

Chapter 7

7. Conclusion

Previous studies [27][28][45][46] have shown that pathway comparison is able to obtain biologically relevant results. In this thesis, a quantitative framework for comparing biochemical pathways was developed. The goal of this framework is to capture and compare the molecular machinery aspect of biochemical pathways. This framework compares biochemical pathways at a very high (abstract) level because detail information such as kinetics data is not readily available for most common pathways.

In this thesis, two models for representing biochemical pathway were developed. First, biochemical pathways were represented as strings of protein names. Next, deterministic finite automata, restricted to have no "loops", were used to represent pathways.

One major contribution of this thesis is that it has shown that ontology of biological terms (e.g. protein names) can be used as a similarity score-calculation criterion for comparing biological networks. This thesis used the NCI Thesaurus, an ontology of protein names commonly used in cancer research, to calculate the similarity matrix for pathway comparison.

Pathway comparison analyses done in this thesis did not reveal any new biological insights. However, the results from the analyses were consistent with current literature in biology. This shows that the pathway representation models and comparison methods used in this thesis were able to produce biologically relevant results. Further pathway comparison experiments will be needed to show more insightful results.

The biochemical pathway representation model can be improved to model biochemical pathways in a more realistic manner. However, one must find a balance between modeling "everything" extensively and keeping the model simple so that pathway representation and comparison can be done in a practical manner.

Bibliography

[1] Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J. D. 1994. *Molecular Biology of The Cell Third Edition.* Garland Publishing, Inc.

[2] Barabasi, A. and Oltvai, Z. N. 2004. Network Biology: Understanding the Cell's Functional Organization. *Nature Reviews Genetics.* 5: 101-113

[3] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C. and Eddy, S. R. 2004. The Pfam protein families database. *Nucleic Acids Research.* 32 (D): 138-141.

[4] Baxevanis, A. D. and Ouellette, B. F. F. 2001. *Bioinformatics.* John Wiley & Sons, Inc.

[5] Cormen, T. H., Leiserson, C. E. and Rivest, R. L. 1990. *Introduction to Algorithms.* The MIT Press.

[6] Covitz, P. A., Hartel, F., Schaefer, C., De Coronado, S., Fragoso, G., Sahni, H., Gustafson, S. and Buetow K. H. 2003. caCORE: A common infrastructure for cancer informatics. *Bioinformatics.* 19: 2402-2412.

[7] Dandekar, T., Schuster, S., Snel, B., Huynen, M. and Bork, P. 1999. Pathway Alignment: Application to the Comparative Analysis of Glycolytic Enzymes. *Journal of Biochemistry.* 343: 115-124.

[8] de Hoon, M. J. L., Imoto, S., Nolan, J. and Miyano, S. Open Source Clustering Software. 2004. *Bioinformatics.* 20(9): 1453-1454.

[9] Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. 1998. *Biological Sequence Analysis.* Cambridge University Press.

[10] Eisen, M. B., Spellman, P. T., Brown P. O. and Bostein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *PNAS.* 95(25) 14863-14868.

[11] Everitt, B. S., Landau, S. and Leese, M. 2001. *Cluster Analysis,* 4th edition. Edward Arnold.

[12] Feng, D. -F. and Doolittle, R. F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution.* 25:351-360.

[13] Fitch, W. M. and Margoliash, E. 1967. Construction of phylogenetic trees. *Science.* 155:279-284.

[14] Galperin, M. Y. 2005. The Molecular Biology Database Collection: 2005 update. *Nucleic Acids Research*. 33: D5-D24.

[15] Gibas, C. and Jambeck, P. 2001. *Bioinformatics Computer Skills.* O'Reilly & Associates, Inc.

[16] Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J. and Parsia, B. 2003. The National Cancer Institute's Thesaurus and Ontology. *Journal of Web Semantics*. 1: 75-80.

[17] Goldsby, R. A., Kindt, T. J., Osborne, B. A. and Kuby, J. 2003. *Immunology Fifth Edition*. W. H. Freeman and Company.

[18] Gruber, T. R. 1993. A translation approach to portable ontologies. *Knowledge Acquisition*. 5(2): 199-220.

[19] Gupta, S. 2003. Molecular signaling in death receptor and mitochondrial pathways of apoptosis (Review). *International Journal of Oncology*. 22: 15-20.

[20] Haque, S. J. and Williams, R. G. 1998. Signal Transduction in the Interferon System. *Seminars in Oncology*. 25(1): 14-22.

[21] Hengartner, M. O. 2000. The biochemistry of apoptosis. *Nature*. 407: 770-776.

[22] Hirschberg, D. S. 1975. A linear space algorithm for computing maximal common subsequences. *Comm. A.C.M.* 18(6): 341-343.

[23] Hopcroft, J. E., Motwani, R. and Ullman, J. D. 2000. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley.

[24] Horton, H. R., Moran, L. A., Ochs, R. S., Rawn, J. D. and Scrimgeour, K. G. 1996. *Principles of Biochemistry Second Edition.* Prentice-Hall, Inc.

[25] Horvath, C. M. The Jak-STAT Pathway Stimulated by Interferon Gamma. 2004. *Science STKE*. 260:tr8.

[26] Kanehisa, M. and Goto, S. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*. 28: 27-30.

[27] Kelley, B. P., Sharan, R., Karp, R. M., Sittler, T., Root, D. E., Stockwell, B. R. and Ideker, T. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS.* 100(20): 11394-11399.

[28] Kelly, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R. and Ideker, T. 2004. PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Research*, 32: W83-W88.

[29] Kelly-Welch, A. E., Hanson, E. M., Boothby, M. R. and Keegan, A. D. 2003. Interleukin-4 and Interleukin-13 Signaling Connections Maps. *Science*. 300: 1527-1528.

[30] Kimura, M. 1983. *The Neutral Theory of Molecular Evolution.* Cambridge University Press.

[31] Knudsen S. 2002. *A biologist's guide to analysis of DNA microarray data.* John Wiley & Sons, INC., New York.

[32] Lehninger, A. L., Nelson, D. L. and Cox M. M. 1993. *Principles of Biochemistry Second Edition.* Worth Publishers, Inc.

[33] Meier, P., Finch, A. and Evan, G. 2000. Apoptosis in Development. *Nature.* 407: 796-801.

[34] Noy, N. F. and McGuinness, D. L. 2001(March). Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report* KSL-01-05 and *Stanford Medical Informatics Technical Report* SMI-2001-0880.

[35] Perrière, G. and Gouy, M. 1996. WWW-Query: an on-line retrieval system for biological sequence banks. *Biochimie*. 78: 364-369.

[36] Rane, S. G. and Reddy, E. P. 2000. Janus kinases**:** components **of** multiple signaling pathways. *Oncogene*. 19(49): 5662-5679.

[37] Rector, A. 2004. Why use a Classifier? When will it help? And when will it not? Alan Rector (University of Manchester). *7th International Prot ég é Conference.*

[38] Regev, A., Silverman, W., and Shapiro, E. 2001. Representation and Simulation of Biochemical Processes using the Pi-Calculus Process Algebra. *Proceedings of the Pacific Symposium of Biocomputing*. 6: 459–470.

[39] Romero, P. and Karp, P. 2003. PseudoCyc, a pathway-genome database for Pseudomonas aeruginosa. *Journal of Molecular Microbiology and Biotechnology*. 5 (4): 230-239.

[40] Romero, P. R. and Karp, P. D. 2004. Using functional and organizational information to improve genome-wide computational prediction of transcription units on pathway-genome databases. *Bioinformatics*. 20 (5): 709-717.

[41] Saitou, N. and Nei, M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*. 4: 406-425.

[42] Schindler, C. 1999. Cytokines and JAK-STAT Signaling. *Experimental Cell Research*. 253: 7-14.

[43] Salomon, D. 2002. *A Guide to Data Compression Methods.* Springer-Verlag New York, Inc.

[44] Thompson, J. D., Higgins, G. and Gibson, T. J. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research.* 22 (22): 4673-4680.

[45] Tohsato, Y., Matsuda, H. and Hashimoto, A. 2000. An Application of a Pathway Alignment Method to the Analysis of Amino Acid Biosynthesis. *Genome Informatics.* 11: 284-285.

[46] Tohsato, Y., Matsuda, H. and Hashimoto, A. 2000. A Multiple Alignment Algorithm for Metabolic Pathway Analysis using Enzyme Hierarchy. *Proceedings ISMB-2000.* 376-383.

[47] Verma, A., Kambhampati, S., Parmar, S. and Platanias, L. C. 2003. Jak family of kinases in cancer. *Cancer and Metastasis Reviews*. 22: 423-434.

[48] Zumdahl, S. S. 1995. *Chemical Principles.* D. C. Heath and Company.

[49] http://java.sun.com/. Sun Java home page.

[50] http://java.sun.com/products/jdbc/. Sun JDBC page.

[51] http://kbrin.a-bldg.louisville.edu/CECS694/. "Introduction to Bioinformatics" lecture notes from the University of Louisville.

[52] http://nciterms.nci.nih.gov/NCIBrowser/Startup.do. NCI Terminology Browser.

[53] http://obelia.jde.aca.mmu.ac.uk/multivar/ca.htm. Cluster Analysis.

[54] http://www.biocarta.com. BioCarta.

[55] http://www.biocyc.org. BioCyc.

[56] http://www.biopax.org/. Biological Pathways Exchange.

[57] http://www.chem.qmul.ac.uk/iubmb/enzyme/. Enzyme Nomenclature.

[58] http://www.chem.qmul.ac.uk/iubmb/enzyme/history.html. Historical Introduction (Enzyme Nomenclature).

[59] http://www.kegg.org. Kyoto Encyclopedia of Genes and Genomes. (KEGG)

[60] http://www.mysql.com/. MySQL home page.

[61] http://www.r-project.org/. The R Project for Statistical Computing.

[62] http://www.resample.com/xlminer/help/HClst/HClst_intro.htm. Hierarchical Clustering.

[63] http://www.stke.org. Signal Transduction Knowledge Environment. (STKE)

[64] http://www.w3.org/2001/sw/. Semantic Web.

[65] http://www-ksl.stanford.edu/kst/what-is-an-ontology.html. What is an Ontology?

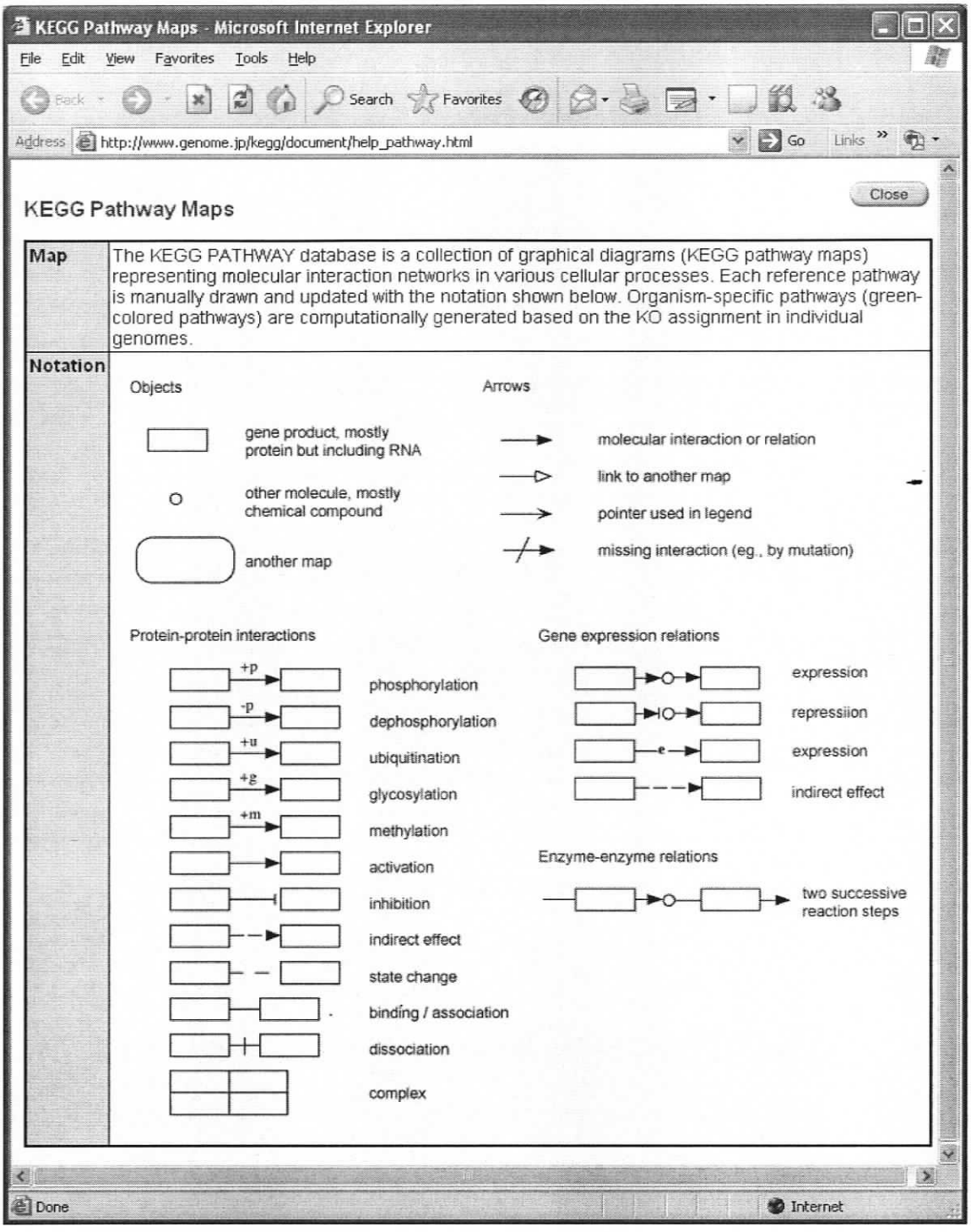
A. NCI Thesaurus Protein Code/Name Table

NCI Protein Code	Protein Name
C16554	Enzyme
C17020	Protein_Tyrosine_Kinase
C17067	Cell_Surface_Receptor
C17678	I-Kappa-B-Alpha_Protein
C17667	Cytokine_Receptor
C17776	FAS_Antigen
C17800	Tumor_Necrosis_Factor_Alpha_Receptor
C17812	TNF_Receptor-Associated_Factor-2
C17907	TRADD_Protein
C17923	Receptor-Interacting_Serine-Threonine_Kinase-1
C18031	Caspase-3
C18153	Caspase
C18174	TRAIL_Receptor-1
C18178	TRAIL_Receptor-2
C18182	Caspase-8
C18287	Caspase-10
C19285	TNF_Receptor_Family_Protein
C19618	Signal_Transducer_and_Activator_of_Transcription
C20027	Protein_Organized_by_Function
C20464	Cytokine
C20494	Interferon_Alpha
C20495	Interferon_Beta
C20496	Interferon_Gamma
C20500	Tumor_Necrosis_Factor_Family_Protein
C20508	Interleukin-4
C20515	Interleukin-13
C20529	FAS_Ligand
C20533	TRAIL_Protein
C20535	Tumor_Necrosis_Factor-Alpha
C26106	Fas-Associated_Via_Death_Domain_Protein
C26231	Adaptor_Signaling_Protein
C26266	Interleukin-2_Receptor_Gamma
C26487	Serine-Threonine_Protein_Kinase-NIK
C28436	Caspase-6
C28439	Caspase-7
C28492	Janus_Kinase-3

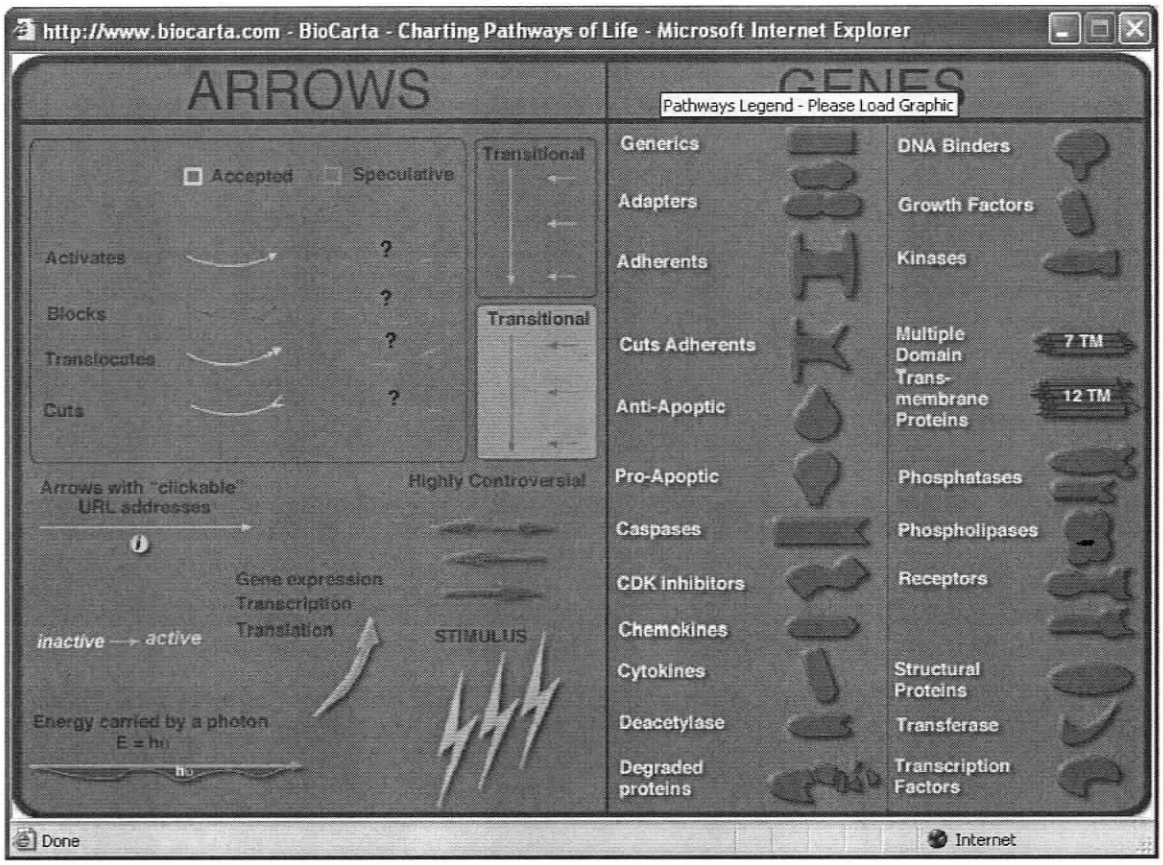
C28493	Janus_Kinase-1
C28494	Janus_Kinase-2
C28659	Signal_Transducer_and_Activator_of_Transcription-1
C28660	Signal_Transducer_and_Activator_of_Transcription-2
C28670	Signal_Transducer_and_Activator_of_Transcription-6
C37278	Interferon_Alpha-Beta_Receptor_Beta_Chain
C37286	Interferon_Gamma_Receptor_Alpha_Chain
C37288	Interferon_Gamma_Receptor_Beta_Chain
MS0001	Tyrosine_Kinase_2
MS0002	Interferon_Regulatory_Factor-9
MS0003	Interleukin-4_Receptor_Alpha
MS0004	Interleukin-13_Receptor_Alpha_1
MS0006	Conserved_Helix-loop-helix_Ubiquitous_Kinase

Table A-1: NCI Thesaurus Protein Code/Name Table.

B. KEGG Diagram Legend



C. BioCarta Diagram Legend



D. Appendix B: Pathway Comparison Software Suit User Manual

D.1. System Requirements

- Java 1.4.2 or higher, http://java.sun.com/ Please note, Java 1.5 is not supported.
- Ant 1.6.2 or higher, http://ant.apache.org/
- MySQL 4.1.7-standard or higher (client and server), http://www.mysql.com/
- Postscript file viewer (required for displaying DFA only) e.g. GSview, http://www.cs.wisc.edu/~ghost/
- R 2.0.0 or higher (required for displaying dendrogram for cluster only), http://www.r-project.org/

D.2. Setup Instructions

D.2.1. Installing the Software

To install the software, please follow the following steps:

1. Copy the jar file to a directory where the software is desired to be installed.
2. Unjar the jar file: `> jar -xf align.jar`
3. Compile source code: `> cd antProject; ant`
4. To setup the postscript file viewer, please edit the "`viewer`" entry in the file `antProject/prefs`. For example, if the postscript file viewer is located at `/usr/bin/gsview`, the entry in `antProject/prefs` should be the following:

```
viewer:/usr/bin/gsview
```

5. Done. To get a list of possible commands: `> ant help`

D.2.2. Setting Up MySQL Tables

To set up the MySQL database and tables, please follow the following steps:

1. Assuming the user is in the directory where the user installed the software (i.e. where `jar -xf align.jar` was performed), change directory:

   `> cd antProject/doc`

2. Log on to MySQL as root: `> mysql -u root`
3. Run the setup script: `mysql> \. mysqlTables`
4. The created MySQL account has username: `align`, and password: `align`. Log off MySQL: `mysql> quit`
5. Import the NCI Thesaurus from NCI: `> ant runMysqlTables`

D.2.3. Updating MySQL Tables

CAUTION: updating the MySQL tables is a nonreversible process, please take extra care. One would need to update the MySQL tables if the protein, to which one wants to refer, is not found in the NCI Thesaurus. Updating the MySQL tables ONLY affects the local database, of course. To undo the changes, one would need to import from the NCI Thesaurus from scratch (i.e. `> ant runMysqlTables`), which means all changes done to the local database would be lost. Please do the following steps to update the MySQL tables.

1. Open the file `antProject/testfiles/updatefile`. If this file does not exist, please create it. The format of this file is described below.
2. For each protein to be added to the database, enter the identifier, name, and any synonyms in `updatefile` in the following format. Please note, the user is

responsible for generating the identifier. Please make sure that the new identifier does not coincide with any existing protein identifier in the database. The format MSXXXX is suggested as it is not the "standard" identifier format used by NCI. One does not need to remove the entries in the `updatefile`, as adding existing entries to the database will be ignored. Therefore, one can keep all changes to the database in the `updatefile`. Currently, the user can only add protein to the local database. The user cannot add any classes of proteins to the database. The following must be added to the `updatefile` as ONE LINE.

```
[identifier];[name];[code of immediate parent];[protein
syn 1];[protein syn 2] ...
```

3. Execute the following command to apply the changes in the `updatefile`:

```
> ant runUpdateTables
```

D.3. Input Files

D.3.1. Format of Input Files

There are two types of input files, one for the linear pathways and one for DFA's without cycles.

1. Linear pathways. The following is an example input file. Please note: the first line (`#LINEAR PATHWAY`) is essential.

```
#LINEAR PATHWAY
## CD95 pathway ################
Tumor Necrosis Factor Ligand Superfamily Member 6
Tumor Necrosis Factor Receptor Superfamily, Member 6
Fas-Associated Via Death Domain
Caspase-8
Caspase-3
```

2. DFA without cycles. The following is an example input file. Please note the following:

   - The first line (`#DFA`) is essential.
   - All lines that starts with "`@`" are part of the DFA definition.
   - State names should not start with capital letters.
   - The following example input file represents the following DFA.

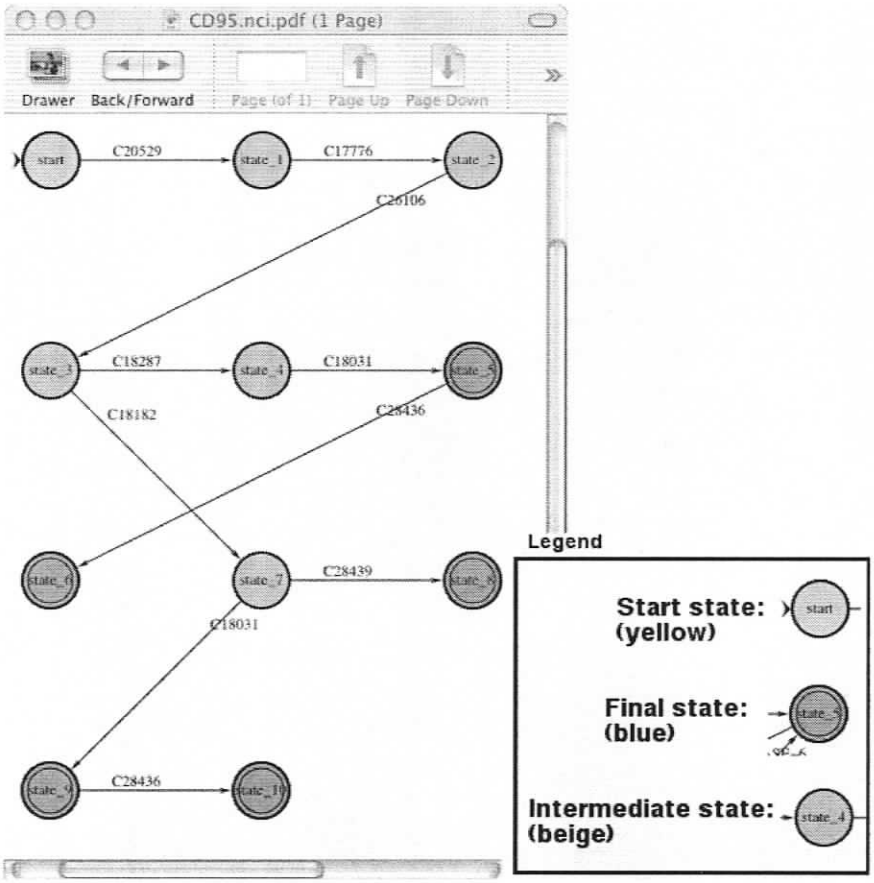


Figure D-1: DFA without cycle representing CD95 signaling pathway.

```
#DFA
## CD95 pathway ################################
## Reference:
##   www.kegg.org
@state names
@start
@state_1
@state_2
@state_3
```

```
@state_4
@state_5
@state_6
@state_7
@state_8
@state_9
@state_10
@end of state names
#-----------------------------------------------
@input symbols
Tumor Necrosis Factor Ligand Superfamily Member 6
Tumor Necrosis Factor Receptor Superfamily, Member 6
Fas-Associated Via Death Domain
Caspase-3
Caspase-6
Caspase-7
Caspase-8
Caspase-10
@end of input symbols
#-----------------------------------------------
@transition table
#------
@start
Tumor Necrosis Factor Ligand Superfamily Member 6
@state_1
#------
@state_1
Tumor Necrosis Factor Receptor Superfamily, Member 6
@state_2
#------
@state_2
Fas-Associated Via Death Domain
@state_3
#------
@state_3
Caspase-10
@state_4
#------
@state_4
Caspase-3
@state_5
#------
@state_5
Caspase-6
@state_6
#------
@state_3
Caspase-8
@state_7
#------
@state_7
Caspase-7
@state_8
#------
@state_7
Caspase-3
@state_9
```

```
#------
@state_9
Caspase-6
@state_10
#------
@end of transition table
#-------------------------------------------
@start state
@start
#-------------------------------------------
@final states
@state_5
@state_6
@state_8
@state_9
@state_10
@end of final states
```

Figure D-2: Example DFA input file.

D.3.2. Preprocess Input Files

One would need to preprocess the input files. The main purpose of this step is to find out if the protein names (in the case of linear pathway) or the input symbol names (in the case of DFA) exist in the local version of the NCI Thesaurus (i.e. the one generated by D.2.2 and D.2.3). The preprocessing software generates input files that can be read by the pathway comparison software. The files generated by the preprocessing software has an ".nci" extension. The user must not modify the .nci files – doing so might result in the pathway comparison software not being able to parse the user-modified .nci files. To preprocess an input file (e.g. assume the input file name is `CD95_1`), do the following:

1. Execute the following: `> ant runMatchNode CD95_1`
2. A file `CD95_1.nci` will be generated in the same directory as `CD95_1`. Examine this file and make sure that all protein names/input symbols entries are in the following format:

`[NCI Protein Code] [Input Protein Name / Input Symbol]`

For example,

```
C20529 FAS_Ligand
```

Please note that the protein names/input symbols in the .nci file will be changed to the identifier name in the NCI Thesaurus. The user would need to make sure that this name refers to the same protein as specified in the original input file. Some synonyms used in the literature might not be used in the NCI Thesaurus. For example, some literature refers to "CD95" as "FAS". However, in the NCI Thesaurus, "FAS" refers to "Fatty Acid Synthase", while "FAS Antigen" refers to "CD95".

If any protein names/input symbols entries in the .nci file is in either one of the following two formats, some manual intervention is required.

```
AMBIGUOUS PROTEIN (jak):
   C28492 Janus_Kinase-3
   C28493 Janus_Kinase-1
   C28494 Janus_Kinase-2
```

The entries above represent the case where the user had entered an ambiguous synonym for a protein the user wants to refer to. The user would need to modify the original input file such that the synonym provided would not be ambiguous. For example, if the user wanted to refer to Janus Kinase-1, the user should use the synonym "JAK1" in the original input file. After the user modified the input file, the user must rerun the preprocessing software (i.e. `> ant runMatchNode`) to generate the .nci files.

```
UNKNOWN PROTEIN: jak9
```

The above entry represents the case where the user had entered a synonym for a protein that is not found in the NCI Thesaurus. The user would first need to make sure that the protein is not in the NCI Thesaurus. It is likely that the synonym of the

protein is not in the NCI Thesaurus but the actual protein is recorded in the NCI Thesaurus under a different name/synonym. Please go to the NCI Thesaurus web interface (http://nciterms.nci.nih.gov/NCIBrowser/Startup.do) to make sure that the unknown protein is in fact not found in the NCI Thesaurus. If the protein is not found in the NCI Thesaurus, the user can add the protein to the local NCI Thesaurus (MySQL) database. To do so, please refer to D.2.3.

D.4. Pathway Comparison

The section describes examples of doing pathway comparison. All operations described below assume that the user is in the directory `antProject` – the top directory of the installed software.

D.4.1. Pair-wise Linear Pathway Alignment

1. Two input files:

Input file 1: `testLINEAR/CD95_1`

```
#LINEAR PATHWAY
## CD95 pathway ################
Tumor Necrosis Factor Ligand Superfamily Member 6
Tumor Necrosis Factor Receptor Superfamily, Member 6
Fas-Associated Via Death Domain
Caspase-8
Caspase-3
```

Input file 2: `testLINEAR/IL4_JAK_STAT_1`

```
#LINEAR PATHWAY
## IL 4 pathway with JAK/STAT pathway
Interleukin-4
Interleukin-4 Receptor Alpha
common cytokine receptor gamma chain
JAK1
JAK3
STAT6
STAT6
```

Figure D-3: Two linear pathway input files.

2. Preprocess input files.

Edit the file `build.xml`: append the input file names in the following entry as shown.

```
<target name="runMatchNode">
  <java fork="true"
classname="pathwayAlign.matchNode.MatchNode">
  <classpath path="${classpath}:${class}"/>
    <arg line="testLINEAR/IFN_G_1 testLINEAR/IFN_A ...
testLINEAR/CD95_1 testLINEAR/IL4_JAK_STAT_1"/>
  </java>
</target>
```

Execute the application to preprocess the input files.

```
> ant runMatchNode
```

Examine the generated files (`testLINEAR/CD95_1.nci` and `testLINEAR/IL4_JAK_STAT_1.nci`). Make sure that there are no `UNKNOWN` and `AMBIGUOUS` entries in the input files.

3. Do pair-wise linear pathway alignment.

Edit the file `build.xml`: add or edit the following entry as shown.

```
<target name="runAlignPathways">
  <java fork="true"
classname="pathwayAlign.align.AlignPathways">
    <classpath path="${classpath}:${class}"/>
      <arg line="testLINEAR/CD95_1.nci
testLINEAR/IL4_JAK_STAT_1.nci"/>
  </java>
</target>
```

Execute the following command to do pair-wise pathway alignment.

```
> ant runAlignPathways
```

4. Expected Results.

```
> ant runAlignPathways
Buildfile: build.xml

runAlignPathways:
```

```
     [java] The input files are: testLINEAR/CD95_1.nci
   testLINEAR/IL4_JAK_STAT_1.nci
     [java] total num of proteins in database is: 1801
     [java] Running main in AlignPathways.java
     [java] Pathways to align are:
     [java] name = testLINEAR/CD95_1.nci
     [java] sequence = C20529 C17776 C26106 C18182 C18031

     [java] name = testLINEAR/IL4_JAK_STAT_1.nci
     [java] sequence = C20508 MS0003 C26266 C28493 C28492 C28670 C28670

     [java] Alignment score = 17.0
     [java] alignment:
     [java]   aligned pathway: C20529 C17776 C26106 C18182 C18031 -----
   - ------
     [java]   aligned pathway: C20508 MS0003 C26266 C28493 C28492
   C28670 C28670
     [java] consensus pathway: C20464 C17667 C20027 C16554 C16554 -----
   - ------


BUILD SUCCESSFUL
Total time: 3 seconds
```

Figure D-4: Expected result for pair-wise linear pathway alignment.

D.4.2. Cluster Linear Pathways

1. Prepare and preprocess input files for the linear pathways being clustered as described in the above section (D.4.1). Assume the input file names are:

   `testLINEAR/IFN_G_1.nci`, `testLINEAR/IFN_A.nci`,

   `testLINEAR/IFN_B.nci` and `testLINEAR/TNF_ALPHA.nci`.

2. Cluster the input pathways.

   Edit the file `build.xml`: add or edit the following entry as shown.

```
<target name="runClusterPathways">
  <java fork="true"
classname="pathwayAlign.align.ClusterPathways">
    <classpath path="${classpath}:${class}"/>
    <arg line="testLINEAR/IFN_G_1.nci testLINEAR/IFN_A.nci
LINEAR/IFN_B.nci testLINEAR/TNF_ALPHA.nci"/>
  </java>
</target>
```

   Execute the following command to cluster the input linear pathways.

```
> ant runClusterPathways
```

3. Expected results.

```
> ant runClusterPathways
Buildfile: build.xml

runClusterPathways:
     [java] The input files are: testLINEAR/IFN_G_1.nci
testLINEAR/IFN_A.nci testLINEAR/IFN_B.nci testLINEAR/TNF_ALPHA.nci
     [java] number of pathways = 4
...
     [java] distanceMatrix is:
     [java] 0 65 66 225
     [java] 65 0 2 244
     [java] 66 2 0 245
     [java] 225 244 245 0
     [java] clusterOrder is:
     [java] 1 0 3
     [java] 2 1;2 0;1;2

BUILD SUCCESSFUL
Total time: 11 seconds
```

Figure D-5: Expected results for clustering linear pathways.

4. One can generate the following script file based on the above results for R to generate a dendrogram.

```
pathwayAlign <- c(65, 66, 225, 2, 244, 245);
attributes(pathwayAlign) <- list(Size = 4, diag=TRUE);
class(pathwayAlign) <- "dist";
names(pathwayAlign) <- c("IFN_G_1", "IFN_A", "IFN_B", "TNF_ALPHA");
pathwayAlign;
str(upgma <- hclust(pathwayAlign, method = "average"));
plot(upgma, hang = -1);
```

Figure D-6: Script file for R to generate a dendrogram.

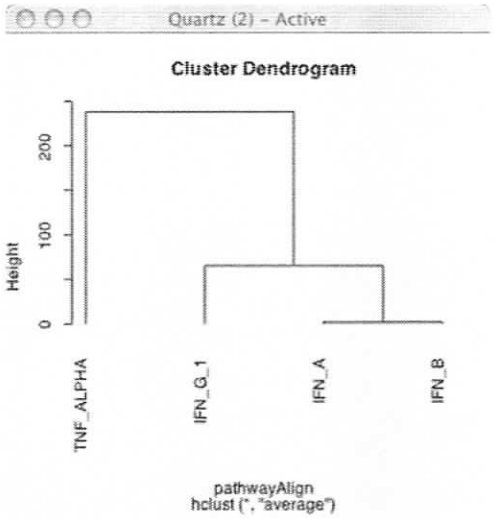


Figure D-7: Dendrogram generated by R.

D.4.3. Multiple Linear Pathway Alignment

1. Prepare and preprocess input files for the linear pathways being aligned as described in the above section (D.4.1). Assume the input file names are:

   `testLINEAR/IFN_G_1.nci`, `testLINEAR/IFN_A.nci`,

   `testLINEAR/IFN_B.nci` and `testLINEAR/TNF_ALPHA.nci`.

2. Do multiple linear pathway alignment.

   Edit the file `build.xml`: add or edit the following entry as shown.

```
<target name="runMultAlignPathways">
  <java fork="true"
classname="pathwayAlign.align.MultAlignPathways">
    <classpath path="${classpath}:${class}"/>
    <arg line="testLINEAR/IFN_G_1.nci testLINEAR/IFN_A.nci
testLINEAR/IFN_B.nci testLINEAR/TNF_ALPHA.nci" />
  </java>
</target>
```

   Execute the following command to cluster the input linear pathways.

```
> ant runMultAlignPathways
```

3. Expected results.

```
% ant runMultAlignPathways
Buildfile: build.xml

runMultAlignPathways:
     [java] The input files are: testLINEAR/IFN_G_1.nci
   testLINEAR/IFN_A.nci testLINEAR/IFN_B.nci testLINEAR/TNF_ALPHA.nci
     [java] testLINEAR/IFN_G_1.nci: C20496 C37286 C37288 C28493 C28494
   C28659 C28659
     [java] testLINEAR/IFN_A.nci: C20494 C37278 C37278 MS0001 C28493
   C28659 C28660 MS0002
     [java] testLINEAR/IFN_B.nci: C20495 C37278 C37278 MS0001 C28493
   C28659 C28660 MS0002
     [java] testLINEAR/TNF_ALPHA.nci: C20535 C17800 C17907 C26106
   C18182 C18031
...
     [java] number of pathways = 4
     [java] testLINEAR/IFN_G_1.nci: C20496 C37286 ------ C37288 C28493
   C28494 C28659 C28659 ------
     [java] testLINEAR/IFN_A.nci: C20494 C37278 ------ C37278 MS0001
   C28493 C28659 C28660 MS0002
     [java] testLINEAR/IFN_B.nci: C20495 C37278 ------ C37278 MS0001
   C28493 C28659 C28660 MS0002
     [java] testLINEAR/TNF_ALPHA.nci: C20535 C17800 C17907 C26106
   C18182 C18031 ------ ------ ------


     [java] Alignment score = -1.0
     [java] alignment:
     [java]   aligned pathway: C20496 C37286 gggggg C37288 C28493
   C28494 C28659 C28659 gggggg
     [java]   aligned pathway: C20494 C37278 gggggg C37278 MS0001
   C28493 C28659 C28660 MS0002
     [java]   aligned pathway: C20495 C37278 gggggg C37278 MS0001
   C28493 C28659 C28660 MS0002
     [java]   aligned pathway: C20535 C17800 C17907 C26106 C18182
   C18031 gggggg gggggg gggggg
     [java] consensus pathway: C20464 C17667 xxxxxx C20027 C16554
   C16554 xxxxxx xxxxxx xxxxxx


BUILD SUCCESSFUL
Total time: 8 seconds
```

Figure D-8: Expect results for multiple linear pathway alignment.

D.4.4. Pair-wise DFA Alignment

1. Two input files for pathways expressed as DFA without cycles.

Input file 1: `testDFA/CD95`

```
#DFA
## CD95 pathway ################################
## Reference:
##   www.kegg.org
@state names
```

```
@start
@state_1
@state_2
@state_3
@state_4
@state_5
@state_6
@state_7
@state_8
@state_9
@state_10
@end of state names
#-----------------------------------------------
@input symbols
Tumor Necrosis Factor Ligand Superfamily Member 6
Tumor Necrosis Factor Receptor Superfamily, Member 6
Fas-Associated Via Death Domain
Caspase-3
Caspase-6
Caspase-7
Caspase-8
Caspase-10
@end of input symbols
#-----------------------------------------------
@transition table
#------
@start
Tumor Necrosis Factor Ligand Superfamily Member 6
@state_1
#------
@state_1
Tumor Necrosis Factor Receptor Superfamily, Member 6
@state_2
#------
@state_2
Fas-Associated Via Death Domain
@state_3
#------
@state_3
Caspase-10
@state_4
#------
@state_4
Caspase-3
@state_5
#------
@state_5
Caspase-6
@state_6
#------
@state_3
Caspase-8
@state_7
#------
@state_7
Caspase-7
@state_8
```

```
#------
@state_7
Caspase-3
@state_9
#------
@state_9
Caspase-6
@state_10
#------
@end of transition table
#-----------------------------------------------
@start state
@start
#-----------------------------------------------
@final states
@state_5
@state_6
@state_8
@state_9
@state_10
@end of final states
```

Input file 2:testDFA/TNF_ALPHA

```
#DFA
## TNF alpha pathway ########################################
## Reference:
##    www.kegg.org
@state names
@start
@state_1
@state_2
@state_3
@state_4
@state_5
@state_6
@state_7
@state_8
@state_9
@state_10
@state_11
@state_12
@state_13
@state_14
@state_15
@state_16
@end of state names
#-----------------------------------------------
@input symbols
Tumor Necrosis Factor-Alpha
Tumor Necrosis Factor Receptor Superfamily, Member 1A
TRADD Protein
Fas-Associated Via Death Domain
Caspase-10
Caspase-8
Caspase-7
Caspase-6
Caspase-3
```

```
RIP
TRAF2
NIK
IKK
I-Kappa-B-Alpha_Protein
@end of input symbols
#-------------------------------------------------
@transition table
#------
@start
Tumor Necrosis Factor-Alpha
@state_1
#------
@state_1
Tumor Necrosis Factor Receptor Superfamily, Member 1A
@state_2
#------
@state_2
TRADD Protein
@state_3
#------
@state_3
RIP
@state_4
#------
@state_4
TRAF2
@state_5
#------
@state_5
NIK
@state_6
#------
@state_6
IKK
@state_7
#------
@state_7
I-Kappa-B-Alpha_Protein
@state_8
#------
@state_3
Fas-Associated Via Death Domain
@state_9
#------
@state_9
Caspase-10
@state_10
#------
@state_10
Caspase-3
@state_11
#------
@state_11
Caspase_6
@state_12
#------
```

```
@state_9
Caspase-8
@state_13
#------
@state_13
Caspase-3
@state_14
#------
@state_14
Caspase-6
@state_15
#------
@state_13
Caspase-7
@state_16
#------
@end of transition table
#-------------------------------------------------
@start state
@start
#-------------------------------------------------
@final states
@state_8
@state_11
@state_12
@state_14
@state_15
@state_16
@end of final states
```

Figure D-9: Two DFA input files.

The DFA diagrams for the above two input files are as follows.

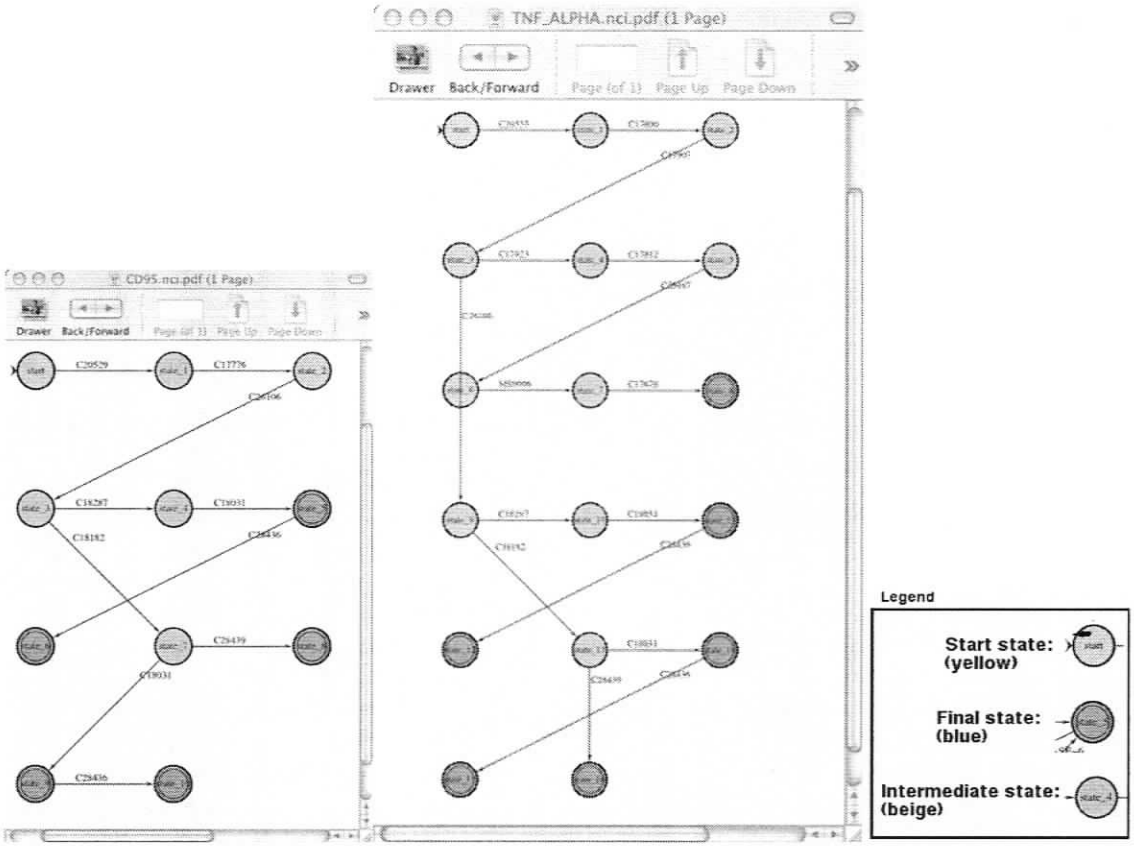


Figure D-10: DFA diagrams for the two DFA input files.

2. Preprocess the input files.

Edit the file `build.xml`: append the input file names in the following entry as shown.

```
<target name="runMatchNode">
  <java fork="true"
classname="pathwayAlign.matchNode.MatchNode">
  <classpath path="${classpath}:${class}"/>
    <arg line="testLINEAR/IFN_G_1 testLINEAR/IFN_A ...
testDFA/CD95 testDFA/TNF_ALPHA"/>
  </java>
</target>
```

Execute the application to preprocess the input files.

```
> ant runMatchNode
```

Examine the generated files (testDFA/CD95.nci and testDFA/TNF_ALPHA.nci). Make sure that there are no UNKNOWN and AMBIGUOUS entries in the input files.

3. Do pair-wise DFA alignment.

   Edit the file build.xml: add or edit the following entry as shown. Please note, "25" is the threshold distance, i.e. the consensus DFA will be constructed using all alignments with distance score less than "25", the threshold. If the keyword "draw" is changed to "not_draw", the application will not generate the DFA diagram postscript files and will not display them.

```
<target name="runDrawAlignDFAs">
  <java fork="true"
classname="pathwayAlign.automata.AlignDFAs">
    <classpath path="${classpath}:${class}"/>
    <arg line="testDFA/CD95.nci testDFA/TNF_ALPHA.nci 25
draw"/>
  </java>
</target>
```

   Execute the following command to cluster the input linear pathways.

```
> ant runDrawAlignDFAs
```

4. Expected Results.

```
% ant runDrawAlignDFAs
Buildfile: build.xml

runDrawAlignDFAs:
     [java] Running AlignDFAs.java main ...
     [java] The input files are: testDFA/CD95.nci testDFA/TNF_ALPHA.nci
     [java] importing state names ...
     [java] importing input symbols ...
     [java] importing transition table ...
     [java] importing start state ...
     [java] importing final state name ...
     [java] importing state names ...
     [java] importing input symbols ...
     [java] importing transition table ...
     [java] importing start state ...
     [java] importing final state name ...
```

```
     [java] Finished importing all DFA!
...
     [java] trying to align two DFAs ...
     [java] total num of proteins in database is: 1801
     [java] alignment distance score = 127.57465725030019
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 21.445521986222698
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 21.445521986222698
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 21.445521986222698
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 27.348142404127668
     [java] alignment distance score = 21.445521986222698
     [java] numBestAlignments = 4
     [java] Alignment score = 67.0
     [java] alignment:
     [java]   aligned pathway: C20529 C17776 ------ C26106 C18287
C18031 C28436
     [java]   aligned pathway: C20535 C17800 C17907 C26106 C18287
C18031 C28436
     [java] consensus pathway: C20500 C19285 ------ C26106 C18287
C18031 C28436

     [java] Alignment score = 67.0
     [java] alignment:
     [java]   aligned pathway: C20529 C17776 ------ C26106 C18182
C18031 C28436
     [java]   aligned pathway: C20535 C17800 C17907 C26106 C18182
C18031 C28436
     [java] consensus pathway: C20500 C19285 ------ C26106 C18182
C18031 C28436

     [java] Alignment score = 67.0
     [java] alignment:
     [java]   aligned pathway: C20535 C17800 C17907 C26106 C18287
C18031 C28436
     [java]   aligned pathway: C20529 C17776 ------ C26106 C18287
C18031 C28436
     [java] consensus pathway: C20500 C19285 ------ C26106 C18287
C18031 C28436

     [java] Alignment score = 67.0
     [java] alignment:
     [java]   aligned pathway: C20535 C17800 C17907 C26106 C18182
C18031 C28436
     [java]   aligned pathway: C20529 C17776 ------ C26106 C18182
C18031 C28436
     [java] consensus pathway: C20500 C19285 ------ C26106 C18182
C18031 C28436
...
     [java] Alignment info ...
     [java] alignment score is: 574.0
     [java] aligned DFA is ...
     [java] DFA info:
     [java] states:
```

```
     [java]   start; state_1; state_2; state_3; state_4; state_5;
state_6;
     [java] input symbols:
     [java]   C20500; C19285; C20027; C26106; C18287; C18031; C28436;
C18182;
     [java] transition table info:
     [java]   start + C20500 -> state_1
     [java]   state_1 + C19285 -> state_2
     [java]   state_2 + C20027 -> state_2
     [java]   state_2 + C26106 -> state_3
     [java]   state_3 + C18287 -> state_4
     [java]   state_4 + C18031 -> state_5
     [java]   state_5 + C28436 -> state_6
     [java]   state_3 + C18182 -> state_4
     [java] start state: start
     [java]   final states:
     [java]   state_6;
     [java] DrawDFA.java: trying to write to:
testDFA/CD95.nci_TNF_ALPHA.nci.ps
     [java] DrawDFA.java: trying to write to: testDFA/CD95.ncisubDFA.ps
     [java] DrawDFA.java: trying to write to:
testDFA/TNF_ALPHA.ncisubDFA.ps
     [java] DrawDFA.java: trying to write to: testDFA/CD95.nci.ps
     [java] DrawDFA.java: trying to write to: testDFA/TNF_ALPHA.nci.ps

BUILD SUCCESSFUL
Total time: 8 seconds
```

Figure D-11: Expect results (standard output) for pair-wise DFA alignment.

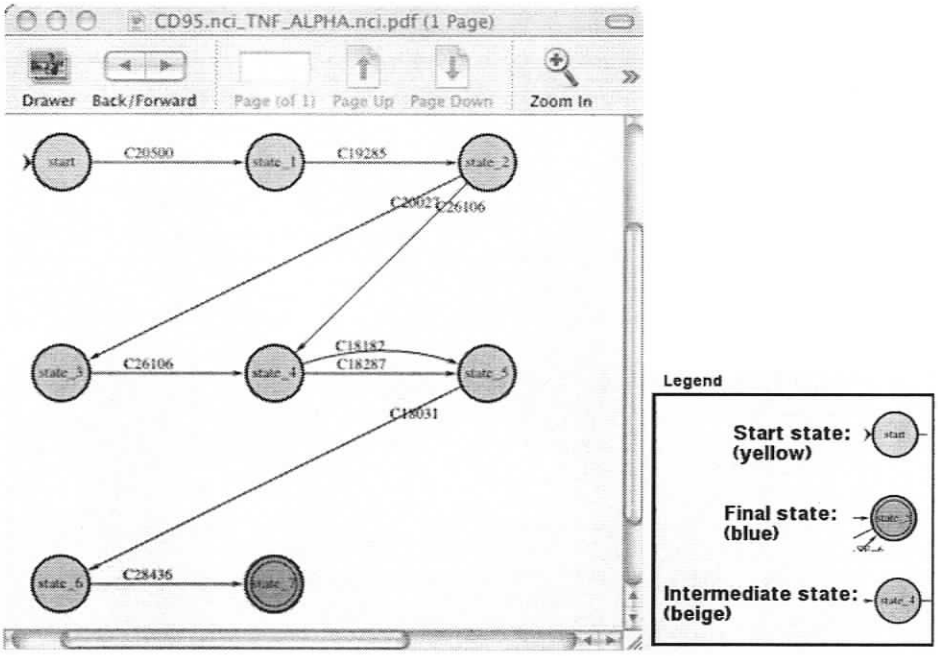


Figure D-12: Consensus DFA.

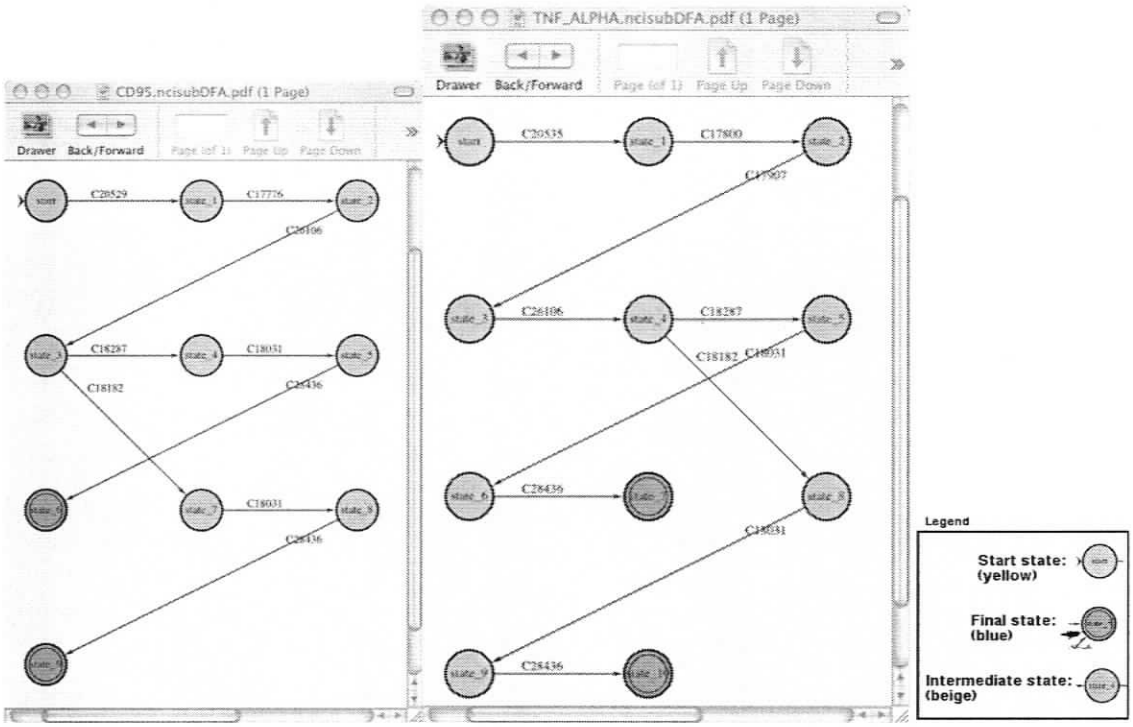


Figure D-13: Parts of the input DFAs that were included in the consensus DFA.

D.4.5. Cluster DFA's

1. Prepare and preprocess input files for the linear pathways being aligned as described in the above section (D.4.4). Assume the input file names are: `testDFA/CD95.nci`, `testDFA/IFN_A.nci`, `testDFA/IL4.nci` and `testDFA/TNF_ALPHA.nci`.

2. Cluster the input DFA's.

   Edit the file `build.xml`: add or edit the following entry as shown.

```
<target name="runClusterDFAs">
  <java fork="true"
classname="pathwayAlign.automata.ClusterDFAs">
    <classpath path="${classpath}:${class}"/>
    <arg line="testDFA/CD95.nci testDFA/IL4.nci
testDFA/IFN_A.nci testDFA/TNF_ALPHA.nci"/>
  </java>
</target>
```

Execute the following command to cluster the input linear pathways.

```
> ant runClusterDFAs
```

3. Expected results.

```
% ant runClusterDFAs
Buildfile: build.xml

runClusterDFAs:
     [java] The input files are: testDFA/CD95.nci testDFA/IL4.nci
testDFA/IFN_A.nci testDFA/TNF_ALPHA.nci
     [java] importing state names ...
     [java] importing input symbols ...
     [java] importing transition table ...
     [java] importing start state ...
     [java] importing final state name ...
     [java] importing state names ...
     [java] importing input symbols ...
     [java] importing transition table ...
     [java] importing start state ...
     [java] importing final state name ...
     [java] importing state names ...
     [java] importing input symbols ...
...
     [java] alignMatrix is:
     [java] 662.0 72.0 33.875 574.0
     [java] 72.0 374.0 107.5 60.0
     [java] 33.875 107.5 200.0 39.375
     [java] 574.0 60.0 39.375 1002.0
     [java] distanceMatrix is:
     [java] 0 258 329 42
     [java] 258 0 112 357
     [java] 329 112 0 349
     [java] 42 357 349 0


BUILD SUCCESSFUL
Total time: 42 seconds
```

Figure D-14: Expected results for clustering DFA's.

4. One can generate a script file based on the above results for R to generate a dendrogram. Please refer to D.4.2 (Cluster Linear Pathways), as this would involve exactly the same operations as generating dendrogram for linear pathway cluster.

E. Appendix C: Pathway Comparison Software Suite, Directory Structure

The following is the directory structure of the pathway comparison software suite.

```
antProject (root)
  |
  +-build.xml - ant build file
  |
  +-class - class files
  |
  +-doc
  |  |
  |  +-architecture - software design
  |  |
  |  +-mysqlTables - SQL statements for setting up MySQL tables
  |
  +-jar - contains jar (library) files
  |
  +-java.policy - policy file for accessing NCI Thesaurus
  |
  +-pathwayAlign
    |
    +-mysqlTables - utility for extracting information from NCI
    |               Thesaurus and store it in mysql tables
    +-matchNode - utility for mapping common protein name to specific
    |             protein code/name in NCI Thesaurus
    +-align - utility for aligning pathways
    |  |
    |  +-AlignPathways.java - align 2 linear pathways using dynamic
    |  |                      programming with no end gaps penalty
    |  +-ClusterPathways.java - cluster linear pathways; returns a
    |  |                        distance matrix
    |  |
    |  +-MultAlignPathways.java - multiple linear pathways alignment
    |  |                          using Feng & Doolittle progressive
    |  |                          alignment
    |  +-RandomAlign.java - generates random linear pathways and align
    |                       them; returns a table of pathway sequence
    |                       lengths vs. alignment scores
    +-automata
    |  |
    |  +-DFA.java - tools for reading DFA's from input files
    |  |
    |  +-DrawDFA.java - generates ps files for input DFA's
    |  |
    |  +-AlignDFAs.java - align 2 DFA's
    |  |
    |  +-ClusterDFAs.java - cluster DFA's; returns a distance matrix
    |
    +-util
    |  |
    |  +-Prefs.java - read the "antProject/pathwayAlign/prefs"
```

```
|                  preference file
|
+-help
   |
   +-Help.java - provides help information
```

Glossary

B lymphocyte (B cell). A lymphocyte, a type of while blood cell, which matures in the bone marrow. It is involved in humoral immune response. Humoral immune response is mediated by antibody present in the plasma, lymph, and tissue fluids. It protects against extracellular bacteria and foreign macromolecules. Transfer of antibodies confers this type of immunity on the recipient. [17]

Cytokine. Low molecular weight intercellular signaling proteins that regulate the intensity and duration of immune response by exerting a variety of effects on lymphocytes and other immune cells. [17]

Janus Kinases (JAKs). A class of enzyme that plays crucial rules in a number of diverse signal transduction pathways that govern cellular survival, proliferation, differentiation and apoptosis. Evidence indicates that JAKs function may integrate components of diverse signaling cascades. [36]

Major histocompatibility complex (MHC). A complex of genes encoding cell-surface molecules that are required for antigen presentation to T cells and for rapid graft rejection. [17]

Signal Transducer and Activator of Transcription (STAT). Cytoplasmic proteins that are activated by phosphorylation in response to a large number of cytokines, growth factors, and hormones. They dimerize and translocate to the nucleus upon activation, where they bind to regulatory cis-elements of specific target genes. STAT proteins have the dual function of signal transduction and activation of transcription. These proteins are activated by phosphorylation on tyrosine in response to different ligands after which they form homodimers or heterodimers that translocate to the cell nucleus where they either directly bind to DNA or act together with other DNA-binding proteins in multiprotein transcription complexes to direct transcription. [52]

T lymphocyte (T cell). A lymphocyte, a type of white blood cell, which matures in the thymus. It is involved in cell-mediated immune response. Cell-mediated immune response is mediated by antigen-specific T cells and various nonspecific cells of the immune system. It protects against intracellular bacteria, viruses, and cancer and is responsible for graft rejection. Transfer of primed T cells confers this type of immunity on the recipient. [17]