

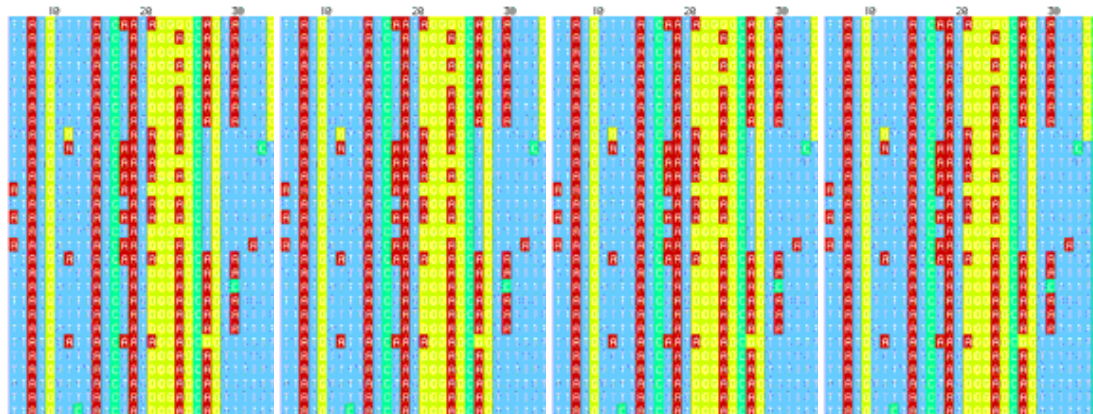
# The Protocol Informatics Project

## Automating Network Protocol Analysis

Marshall Beddoe ([mbeddoe@baselineresearch.net](mailto:mbeddoe@baselineresearch.net))

Copyright © 2004, 2005 Baseline Research

<http://www.baselineresearch.net>



# Before We Start

- HTTP will be used for visualization of concept
  - ◆ Most people know HTTP
  - ◆ PowerPoint slides are only so large
- Questions will be gladly answered at the end

My email: [mbeddoe@baselineresearch.net](mailto:mbeddoe@baselineresearch.net)

PI Homepage: <http://www.baselineresearch.net/PI>

# Objective of Protocol Analysis

- Determine protocol fields
- Understand structure of requests and responses
- Simplified Plaintext Example: HTTP
  - ◆ GET /index.html HTTP/1.0
    - GET: Keyword
    - /index.html: Filename
    - HTTP/1.0: Keyword
- Why is this knowledge important?
  - ◆ Understanding proprietary protocols
  - ◆ Finding vulnerabilities in unknown or badly documented protocols via fuzzing

# Problems with Protocol Analysis

- Binary protocols
- Large amount of data
- Variable length fields
- Time consuming
- Amazingly boring
  
- There must be a better way...
  - ◆ Enter bioinformatics

# Bioinformatics

- What is Bioinformatics?
  - ◆ “The use of mathematical and informational techniques, including statistics, to solve biological problems” - Wikipedia
  - ◆ Processing of large amounts of structured, yet complex data
    - Operates on large sequences of strings to find patterns
  - ◆ Objective: To find genes that produce specific proteins by performing a series of comparisons.
  - ◆ Mapping of phenotypes to genotypes
    - Example: Attached earlobes to the sequence: ATTGAC

# Protocol Analysis & Bioinformatics

- Similarities
  - ◆ Both operate on large sequences of data
  - ◆ Whereas bioinformatics helps find specific genes that produce proteins, protocol analysis finds specific fields in a packet
  - ◆ Both work through a series of compares and contrasts between a large number of samples
- Creating an application that helps understand structured, complex data would be an asset when doing this type of analysis..

# Tech Behind the Talking Points

- Sequence Alignment Fundamentals
  - ◆ Needleman-Wunsch
  - ◆ Smith-Waterman
- Similarity Matrices
  - ◆ BLOSUM, PAM
- Phylogenetic Trees
  - ◆ UPGMA
- Multiple Alignment
  - ◆ Phylogeny

# Sequence Alignment

- Base technology used in bioinformatics
- Idea: Take two sequences regardless of length and align them to each other so both have equal length
- Gaps are inserted when needed to achieve the maximum alignment of the sequences
- Example of DNA alignment:
  - ◆ TCAT---CAA
  - ◆ ||| |
  - ◆ TCATGGGCAA
- Notice the gaps inserted into sequence one to force length alignment
- Simple concept right?



# Needleman-Wunsch Algorithm

- Dynamic programming algorithm
- Performs global alignment on a pair of sequences
  - ◆ Global means that all characters in the sequence participate in the alignment
  - ◆ What goes in, comes out
- Used for analyzing closely related structures

# Dynamic Programming

- Dynamic programming is not coding
- Idea: Break problem into sub-problems
- Operations mainly on matrices
- Results of previous computations are saved and used by the remaining sub-problems
- Needleman Wunsch is a DP algorithm

# How NW Works

- Sequence one is placed in the top-most row and sequence two is placed in the left-most column.
- For each cell, perform the following:
  - ◆ Assign similarity values
  - ◆ Assess possible pathways through matrix (left, up and diagonal), assigning the current cell with value of the maximum scoring pathway using:

$$M_{i,j} = \text{MAX}(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$$

where  $w$  is the gap penalty (currently 0) and  $S$  is the similarly weight

- ◆ Construct a pathway from the highest scoring cell to the beginning of the matrix to get the maximum global alignment
- A gap penalty is used to decrease the preferred number of gaps in the final alignment

# In Other Words: Step One

	G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	1	.	0	
G	1																						
E		1																					
T			1																				
				1											1								
/					1																1		
															1								
H																1							
T			1														1	1					
T			1														1	1					
P																			1				
/					1																1		
1																						1	
.										1													1
0																							1

- Characters that are similar receive a scoring of 1 (for now)

# In Other Words: Step Two

		<b>G</b>	<b>E</b>	<b>T</b>		/	<b>i</b>	<b>n</b>	<b>d</b>	<b>e</b>	<b>x</b>	<b>.</b>	<b>h</b>	<b>t</b>	<b>m</b>	<b>l</b>		<b>H</b>	<b>T</b>	<b>T</b>	<b>P</b>	/	<b>1</b>	<b>.</b>	<b>0</b>	
	<b>0</b>	<b>0</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	<b>0</b>	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>E</b>	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>T</b>	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0
/	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0
<b>H</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0
<b>T</b>	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	<b>1</b>	0	0	0	0	0
<b>T</b>	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	<b>1</b>	0	0	0	0	0
<b>P</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0	0
/	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0
<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0
<b>.</b>	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>	0
<b>0</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>1</b>

Position 1,1

$$M_{i,j} = \text{MAX}(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$$

$$M_{i-1,j-1} + S_{i,j} = 1$$

$$M_{i,j-1} + w = 0$$

$$M_{i-1,j} + w = 0$$

$$w = 0$$

# In Other Words: Step Two

		<b>G</b>	<b>E</b>	<b>T</b>		/	<b>i</b>	<b>n</b>	<b>d</b>	<b>e</b>	<b>x</b>	<b>.</b>	<b>h</b>	<b>t</b>	<b>m</b>	<b>l</b>		<b>H</b>	<b>T</b>	<b>T</b>	<b>P</b>	/	<b>1</b>	<b>.</b>	<b>0</b>	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>E</b>	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>T</b>	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
/	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
<b>H</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
<b>T</b>	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
<b>T</b>	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
<b>P</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
/	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
<b>.</b>	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
<b>0</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Position 1,2

$$M_{i,j} = \text{MAX}(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$$

$$M_{i-1,j-1} + S_{i,j} = 0$$

$$M_{i,j-1} + w = 1$$

$$M_{i-1,j} + w = 0$$

$$w = 0$$

# In Other Words: Step Two

	G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	1	.	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>E</b>	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>T</b>	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
<b>H</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
<b>P</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
<b>1</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
<b>.</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	D
<b>0</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	E

$$M_{i,j} = \text{MAX}(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$$

# In Other Words: Step Three

	G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	1	.	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>E</b>	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>T</b>	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
<b>H</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
<b>P</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
<b>1</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
<b>.</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	D
<b>0</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	E

- Starting in cell with highest value (0xE), look at neighboring cells, choose cell with maximum value and traverse there.



# In Other Words: Step Three

	G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	l	.	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>E</b>	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>T</b>	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
<b>H</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
<b>P</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
<b>l</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
<b>.</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	D
<b>0</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	E

# In Other Words: Step Three

		G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	l	.	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
/	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
H	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
T	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
T	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
P	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
/	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
l	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
.	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	6	7	8	9	A	B	C	D	D
0	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	6	7	8	9	A	B	C	D	E

# What did this do?

- Now that we computed a path through the matrix, we can apply the rules of NW to obtain two aligned sequences
- Anytime the path travels upwards or to the left, a gap is inserted into a sequence
- Upwards affects sequence 1 (row)
- Left affects sequence 2 (column)

# The Result

	G	E	T	/	i	n	d	e	x	.	h	t	m	l	H	T	T	P	/	1	.	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>G</b>	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>E</b>	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>T</b>	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
<b>H</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	7	7	7	7	7	7	7
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	8	8	8	8	8	8
<b>T</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	9	9	9	9	9
<b>P</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	A	A	A	A
<b>/</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	B	B	B
<b>1</b>	0	1	2	3	4	5	5	5	5	5	5	5	5	5	6	7	8	9	A	B	C	C	C
<b>.</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	D
<b>0</b>	0	1	2	3	4	5	5	5	5	5	6	6	6	6	6	7	8	9	A	B	C	D	E

```

GET /index.html HTTP/1.0
|||||           |||||
GET /_____ HTTP/1.0
    
```

# Analyzing the Results

```
GET /index.html HTTP/1.0
|||||           |||||
GET /_____ HTTP/1.0
```

- We can easily discern the protocol fields from these results
  1. GET / is considered a keyword
  2. index.html is subject to deletion (gaps), and is therefore considered a variable length field
  3. Followed by keyword HTTP/1.0

# Smith Waterman Algorithm

- Finds the maximum local alignment
  - ◆ Most similar portions of two sequences
- Especially useful when determining similarity between sequences
- Local alignment is a strong scoring mechanism, whereas global is weak due to the number of gaps that are inserted to satisfy global alignment.
- Uses most of Needleman Wunsch

# How SW Works

- Differences between SW and NW
  - ◆ Positive scores for related chars
  - ◆ Negative scores for mismatches and gaps
  - ◆ Returns when score hits 0
- Gap Penalty = -2
- Match = 2
- Mismatch = -1

# SW in Action

- Aligning
  - ◆ Seq1 = GET /index.html HTTP/1.0
  - ◆ Seq2 = GET / HTTP/1.0
- Results: HTTP/1.0, score = 18
- Distance between seq1 and seq2 =  
 $SW(seq1, seq1) - SW(seq1, seq2) = 48 - 18 = 30$



# How SW is Used

- Smith-Waterman is not used in PI for determining protocol fields
- Instead, it is used to calculate relative distance between sequences for clustering purposes and tree building
- In Bioinformatics, Smith-Waterman is used for multiple sequence alignment because DNA has no defined start and end, whereas a packet does

# Similarity Matrices

- Each character similarity is weighted
- In the earlier NW example, the value of  $S$  was 1
- In Bioinformatics, similarity matrices are used to optimize alignments of sequences.
  - ◆ Markov chain probability table
  - ◆ Based on observed mutations accepted in evolution. Adenine can mutate into thymine, etc.
- Applications to protocol analysis? Datatypes
  - ◆ Binary data mutates into other binary data, as ASCII mutates into other ASCII

# PI Similarity Matrices

- 256x256 matrix
- Contains mutation probabilities between every character
- Direct match has probability of 1
- Others are categorized and weighted
- Arbitrary example:
  - ◆ ASCII character set, probability = .3
  - ◆ ASCII printable, probability = .4
  - ◆ Binary, probability = .4

# What this Allows

- This allows more optimized alignments, with sequences converging on similar data types and reduces the number of incorrect gaps
- Similarity matrices must be tweaked
  - ◆ It is not uncommon to spend a lot of time creating these matrices
  - ◆ Bioinformatics scientists spend years perfecting their version of similarity matrices (BLOSUM, PAM, etc.)

# Example

- Example:
  - ◆ Seq1 = ABCDEF1<sup>b</sup>2<sup>b</sup>3<sup>b</sup>
  - ◆ Seq2 = HIJK4<sup>b</sup>5<sup>b</sup>
- Alignment without proper scoring:
  - ◆ ABCDEF1<sup>b</sup>2<sup>b</sup>3<sup>b</sup>
  - ◆ H \_ I \_ JK \_ 4<sup>b</sup> \_ 5<sup>b</sup>
- Alignment with proper scoring:
  - ◆ ABCDEF1<sup>b</sup>2<sup>b</sup>3<sup>b</sup>
  - ◆ HIJK \_\_\_\_\_ 4<sup>b</sup>5<sup>b</sup>

# What Now?

- Illustrated the ability to align two sequences to each other and discern packet fields
- Shown how similarity matrices can be used to optimize alignment
- Is it really useful only comparing two sequences?

# Multiple Sequence Alignment

- Act of aligning more than 2 sequences
- Uses NW as alignment algorithm
- Uses SW as scoring algorithm
- Computation issues





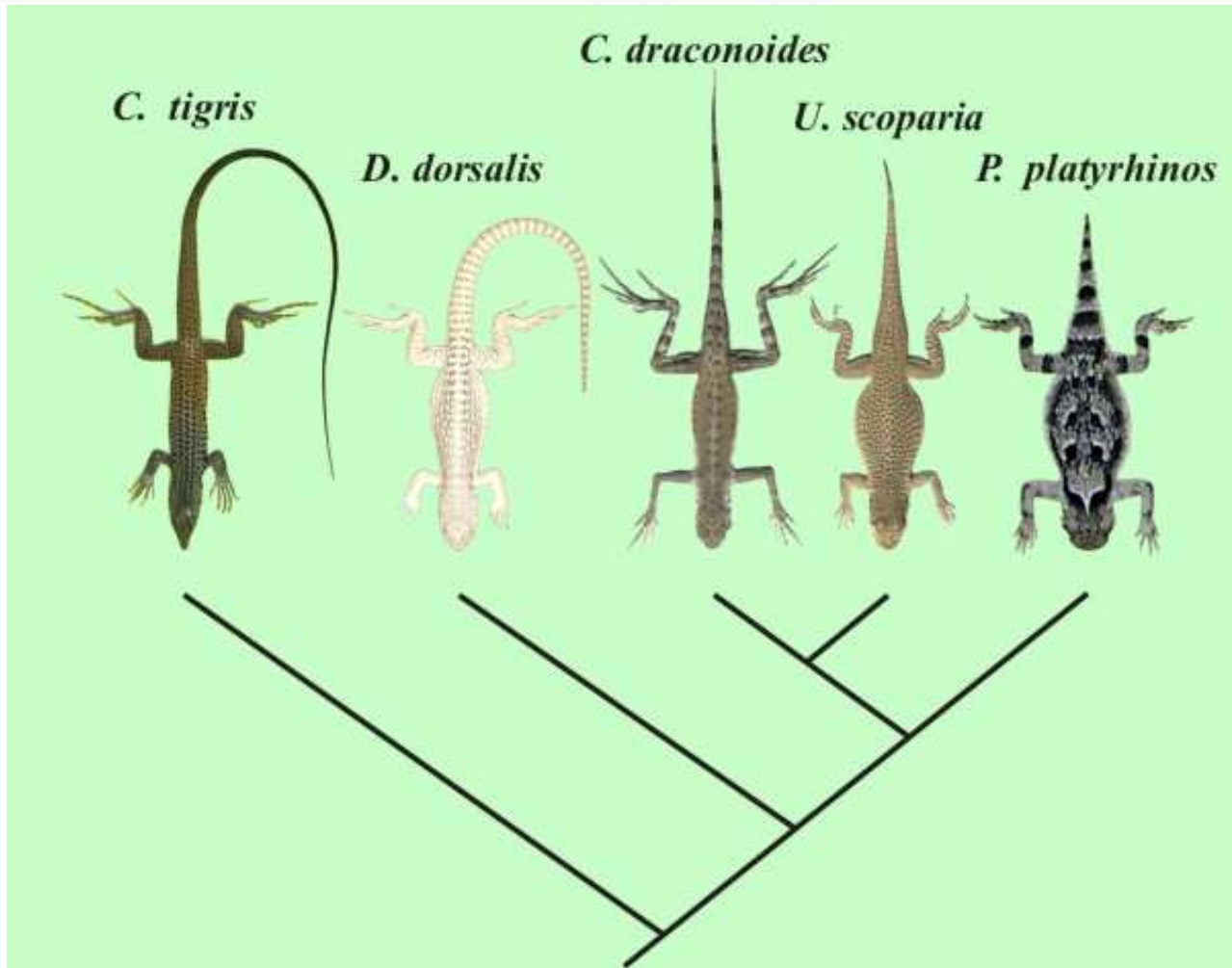
# Heuristic Sequence Alignment

- Sacrificing accuracy for time
- Objective: To align every sequence to each other in a reasonable amount of time
- However, results are never perfect

# Phylogenetic Trees

- A tree of evolutionary development
  - ◆ Used in biology to construct taxonomic groupings based purely on DNA analysis as opposed to fossil records
- Typically binary trees
- Interesting parallel in protocol analysis
  - ◆ A protocol mimics evolution by changing fields
  - ◆ This can be characterized as a mutation
- What came first? GET /index.html or GET / ?

# Phylogeny in Biology



# Creating Phylogenetic Trees

- UPGMA cluster distance algorithm
  - ◆ Unweighted Pair Group Method using Arithmetic Averages

$$d_{i,j} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{p,q}$$

Where  $d_{i,j}$  is the distance between two clusters  $C_i$  and  $C_j$

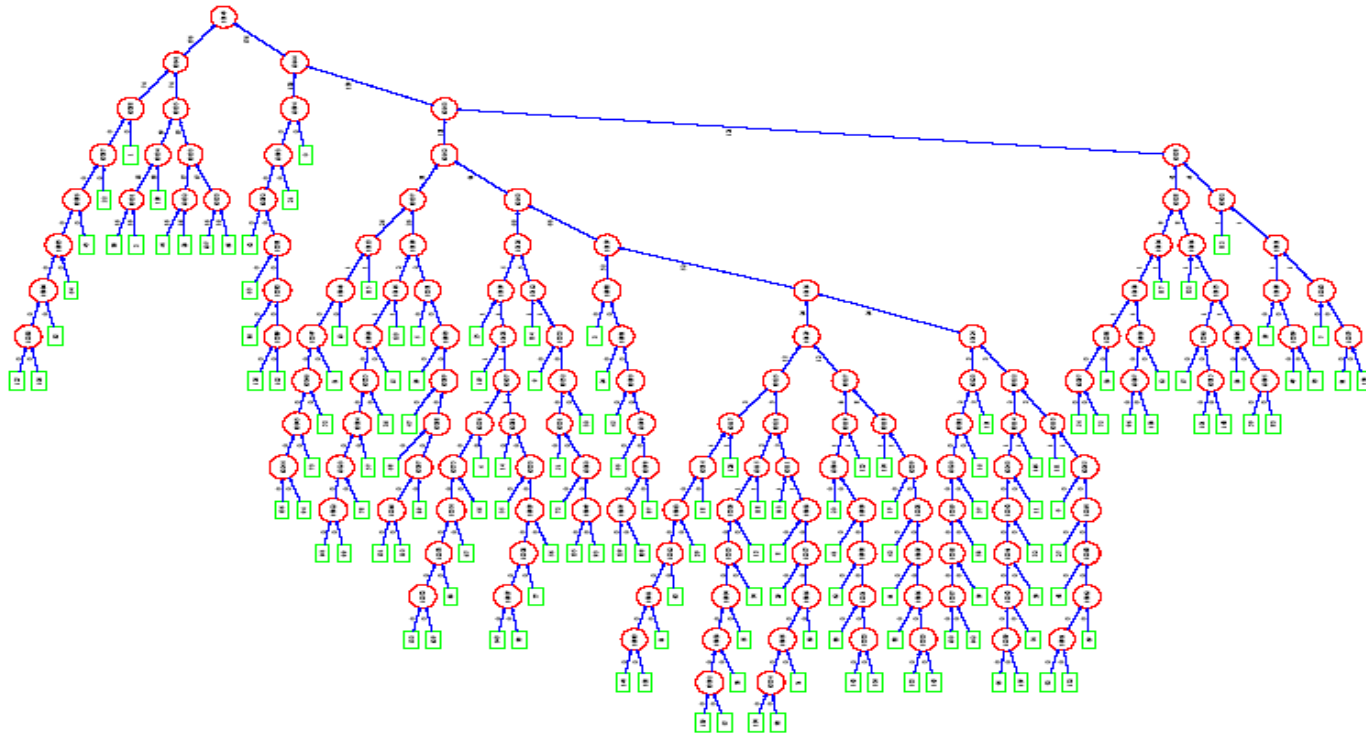
$d_{p,q}$  = Smith Waterman score

# Building the Tree

1. Place each sequence into an individual cluster, insert cluster into universal set
2. Use UPGMA algorithm to calculate distance between each cluster, finding two clusters where  $d_{ij}$  is minimal
3. Create a new cluster  $k$ .  $C_k = C_i \cup C_j$
4. Define a node  $k$  with child nodes  $i$  and  $j$
5. Add  $C_k$  to the universal set and remove  $C_i$  and  $C_j$

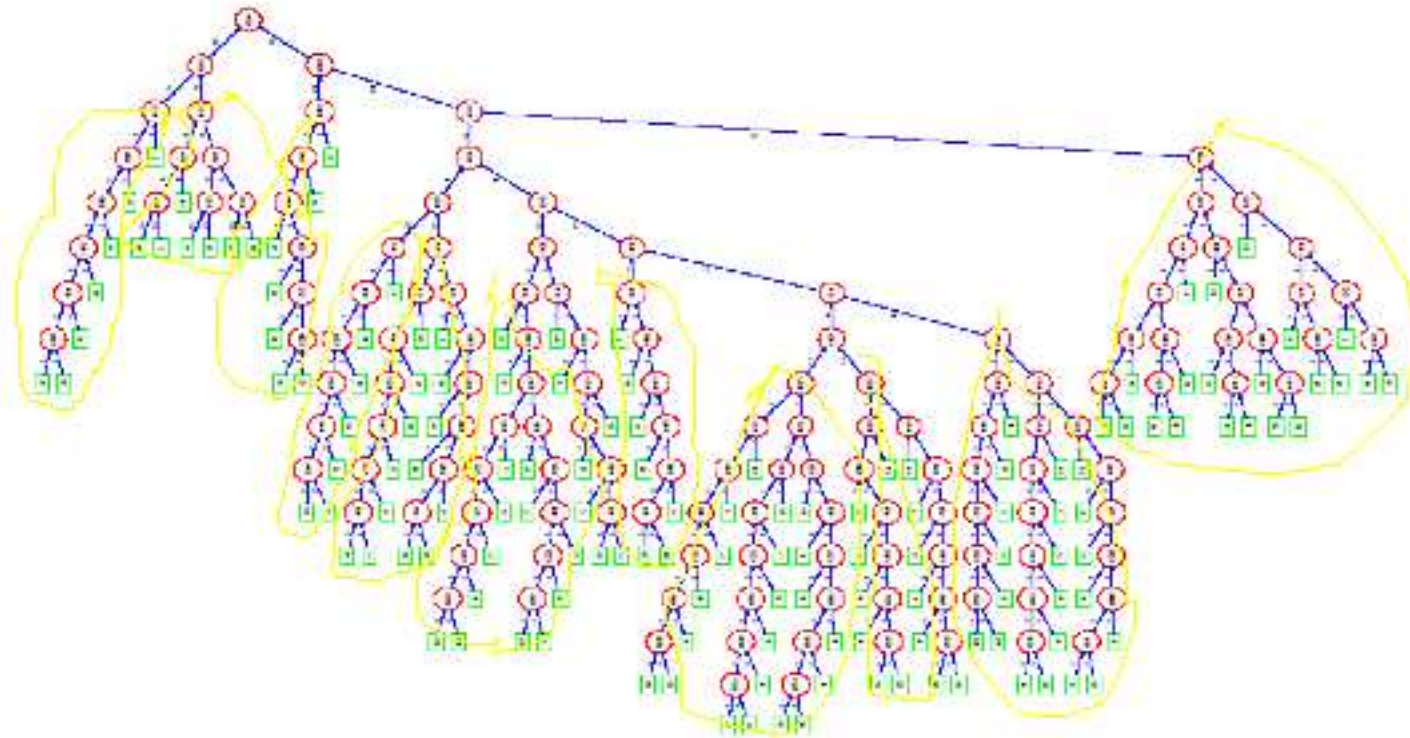
# Phylogeny in Protocol Analysis

## Phylogenetic tree of the SMB protocol



# More Than a Pretty Picture

Phylogenetic tree of the SMB protocol



# The Tree is your Guide

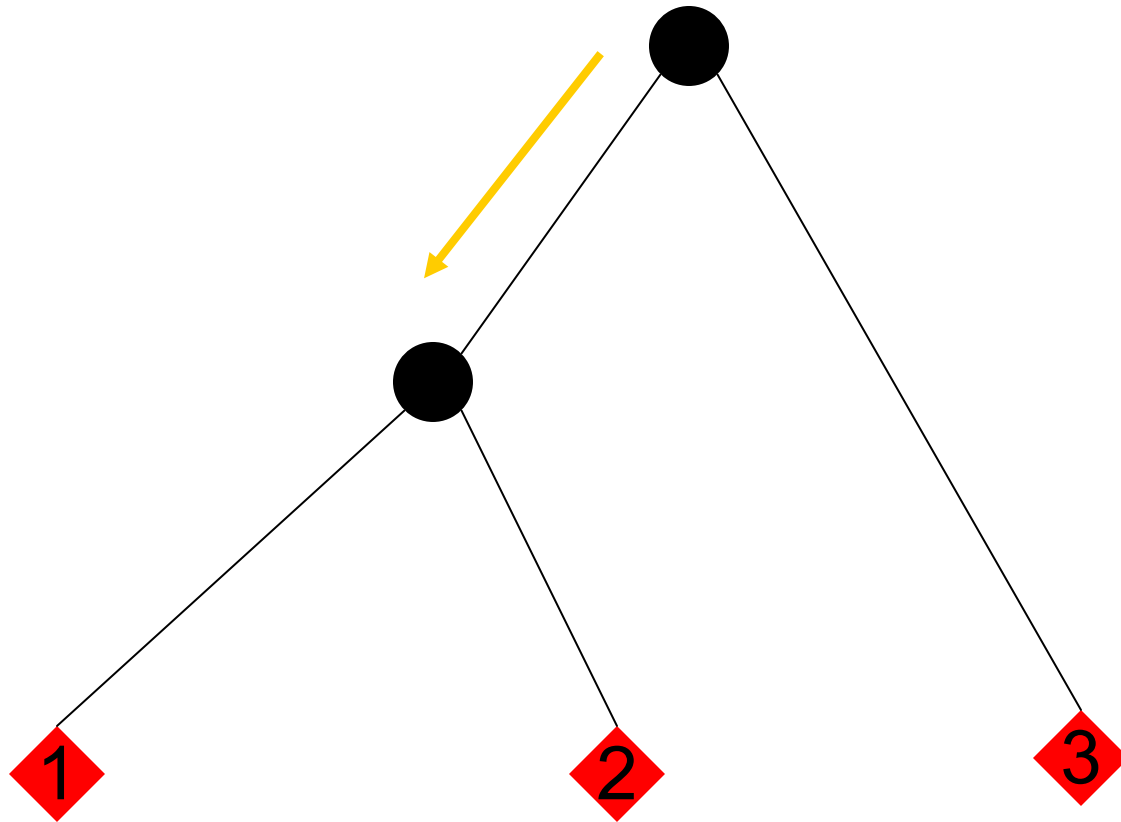
- Helps categorize subtypes of a particular protocol
  - ◆ SMB contains at least 11 subtypes in this tcpdump file
- Tree acts as a guide to perform actual multiple sequence alignment
- As opposed to NP-complete hypercube traversal, the UPGMA tree performs  $n$  comparisons where  $n$  is equal to the depth of the tree minus the root.



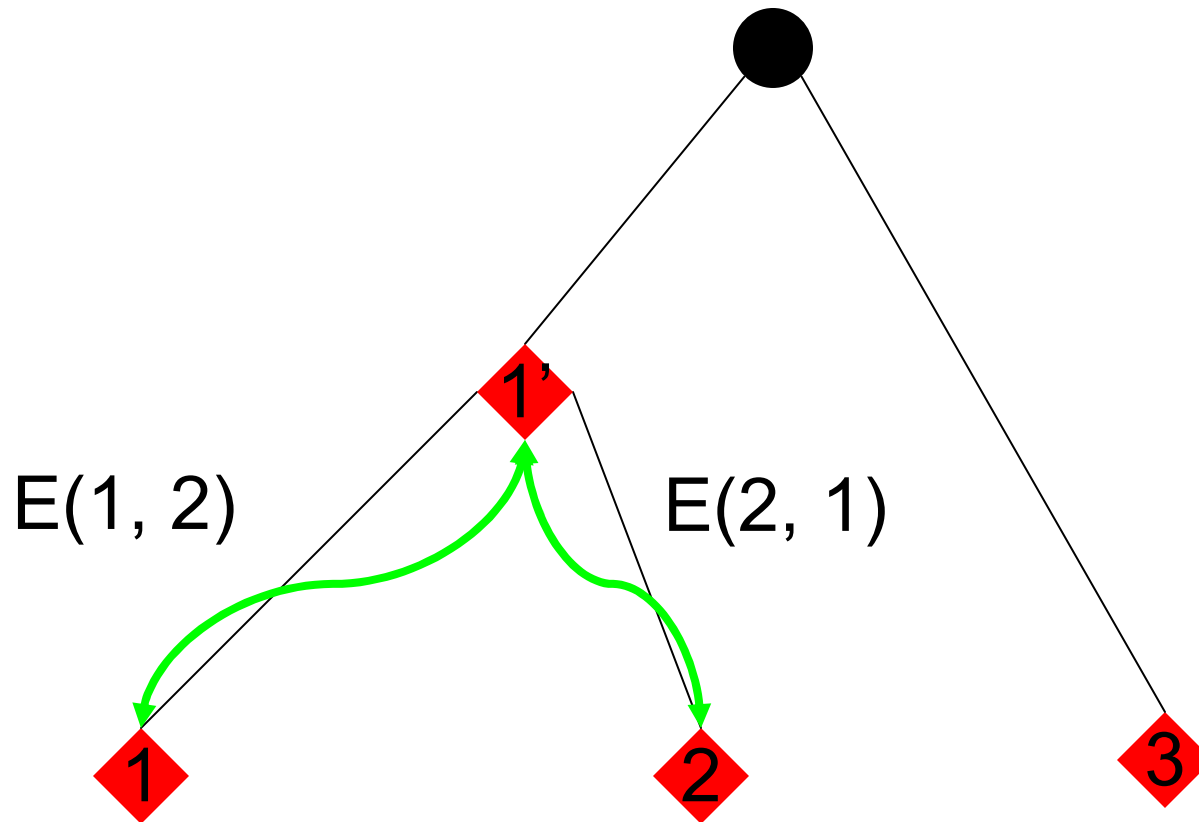
# Multiple Sequence Alignment

- Rule: Once a gap always a gap
- Recursive Traversal Mechanism
  - ◆ If root is NULL, go left, then right
  - ◆ If left is !NULL and right !NULL, align sequences and choose the sequence with the least number of gaps inserted.
    - Seq1: GET /index.html HTTP/1.0
    - Seq2: GET /\_\_\_\_\_ HTTP/1.0
    - Therefore: Seq1 is chosen to be the representative
  - ◆ Place new sequence in root
  - ◆ Keep track of edits in edge

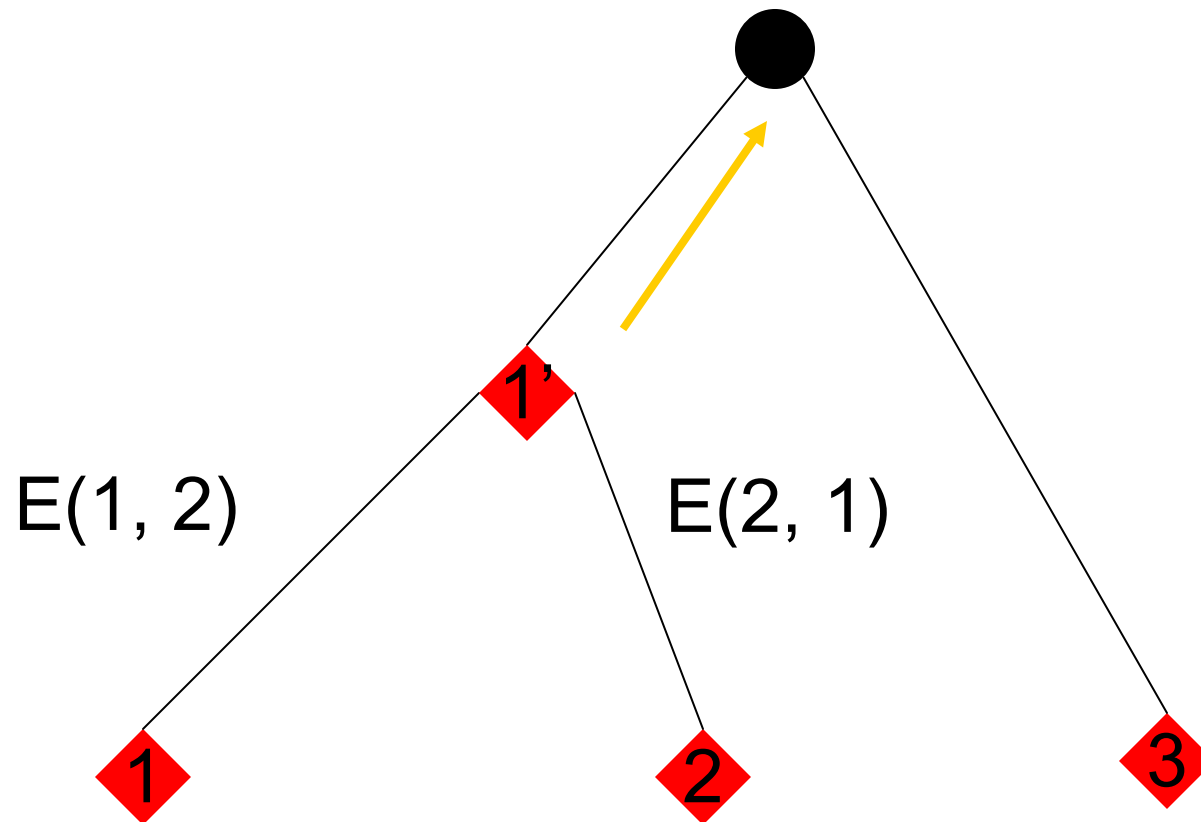
# Tree Traversal Algorithm



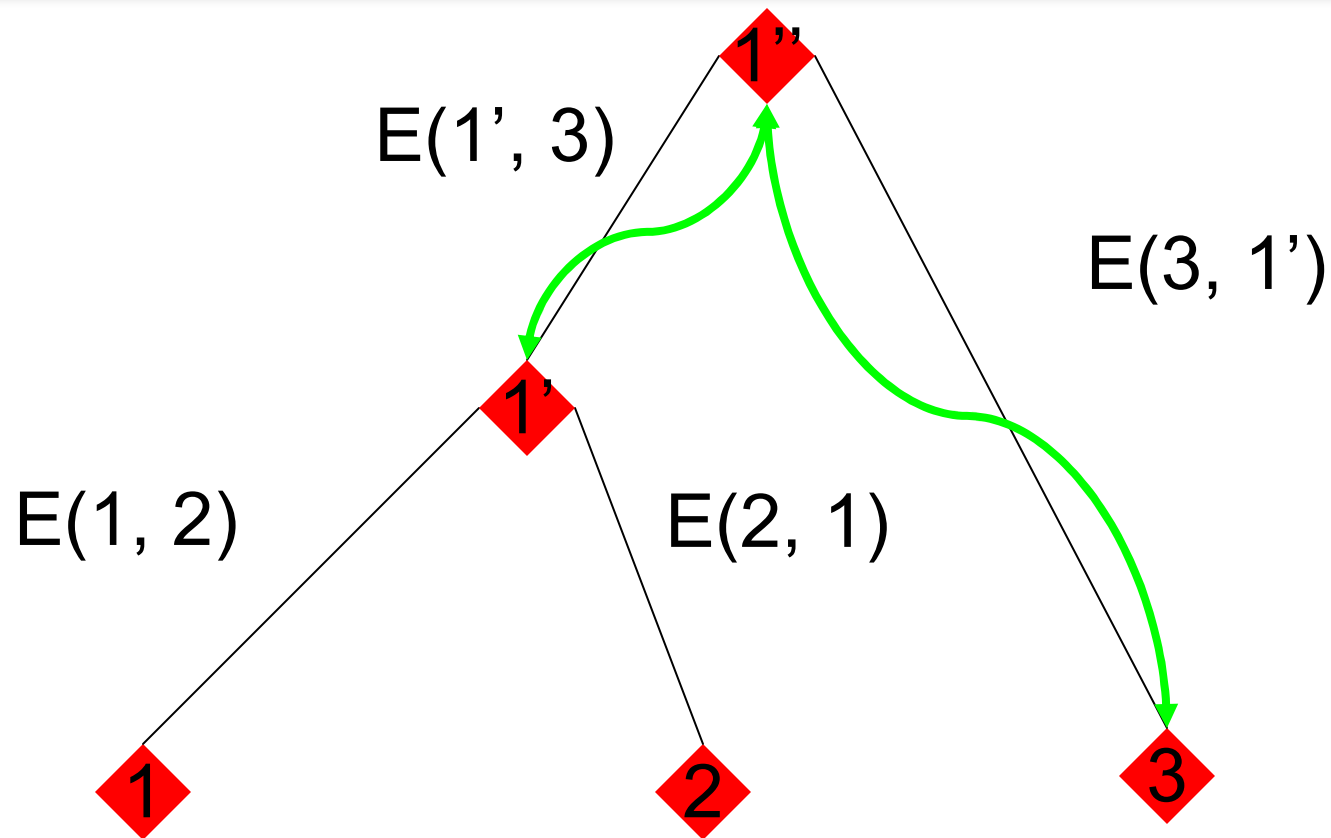
# Tree Traversal Algorithm



# Tree Traversal Algorithm



# Tree Traversal Algorithm



# Therefore

Sequence 1 Aligned =  $E(1,2) + E(1',3)$

Sequence 2 Aligned =  $E(2,1) + E(1',3)$

Sequence 3 Aligned =  $E(3, 1')$

# Analyzing the Results Qualitatively

## Example

```
GET /cgi-bin_/whois.pl HTTP/1.0 Host: _____a__rin.net User-Agent: __Opera__ Accept: text/xml
GET /__i__ndex.h__tml HTTP/1.0 Host: www.yahoo____.com User-Agent: Mozilla/5.0 Accept: text/xml
GET /_____ HTTP/1.0 Host: www.__google.com User-Agent: _____IE4.0 Accept: text/xml

GET /???????????????????? HTTP/1.0 Host: ??????????????.??? User-Agent: ?????????????? Accept: text/xml
```

## Conclusion:

```
GET / <variable> HTTP/1.0 Host: <variable>.<variable> User-Agent: <variable> Accept: text/xml
```

# Analyzing the Results Quantitatively

- Statistical analysis on columns
  - ◆ Histograms
    - Build a consensus sequence as performed on previous
    - Mutation rates & offset comparison
      - Group based on mutation rate and composition
- Beware of junk data
  - ◆ In last example, junk data could have been a POST in a sea of GETs
- Classification is your friend
  - ◆ If you can adequately classify in beginning, data results will be clearer
    - Entropic edit distance, N-gram analysis, local sequence alignment



# Experimental Phase

- Initial thought: Simply separate dynamic data versus static data, however, this is not verbose enough
- Identifying integer fields: Build n-gram frequency tables for 1, 2 and 4 byte window sizes
- Observe rate of mutation for each n-gram
  - ◆ Example 1: If two consecutive bytes mutate at the same rate, chances are they are part of the same field and perhaps a checksum
  - ◆ Example 2: If in two consecutive bytes, the LSB increments faster than the GSB, it may be a 16-bit sequence identifier field.

# Next Steps

- Current Ideas
  - ◆ Building protocol profile based on each consensus and aligning with each sequence individually will breakdown the structure
  - ◆ Build single consensus sequence to describe entire protocol
    - Not usually feasible since many block-based protocols such as ISAKMP, SMB, etc. have many layers.
  - ◆ Present data in an intuitive way to allow improved human estimation and understanding
    - Colors, interface design, etc.
    - This can never be fully automated if accuracy is in mind

# Applications

- This technology can be used for:
  - ◆ Understanding unknown protocols
  - ◆ Fuzz network protocols more efficiently
    - Instead of writing protocol specifications to fuzz against, have them be auto-generated from a tcpdump sample
  - ◆ Learning the structure of any sequence containing complex and somewhat random data
- Do you have any ideas?

# Conclusions

- Never be fully automated 100%
- Experimental technology
- Framework under development
  - ◆ Python/C++, cross platform
  - ◆ Widget based visual programming interface similar to the Orange data mining application (<http://magix.fri.uni-lj.si/orange/>)
  - ◆ Open source and looking for interested people to help
- Closing note: Solutions to computer related problems can be found in other sciences. It is important to expand your horizons.

# Questions/Comments/Ideas?

- Thanks for coming

Marshall Beddoe [mbeddoe@baselineresearch.net](mailto:mbeddoe@baselineresearch.net)

Baseline Research <http://www.baselineresearch.net/PI>

If you are interested in contributing, please contact me.