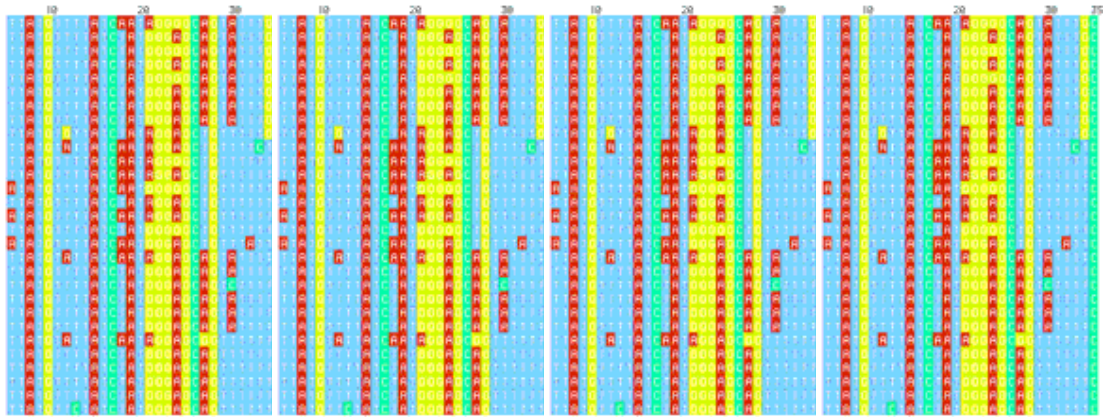
The Protocol Informatics Project

Automating Network Protocol Analysis

Debuted at Toorcon 2004 by
Marshall Beddoe (mbeddoe@baselineresearch.net)

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
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

Problems with Protocol Analysis

- Binary protocols
- Large amount of data
- Dynamically sized 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
  - Needleman–Wunsch
- 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 amino acid 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:

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

    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 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

		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	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	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	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	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	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	5
	0	1	2	3	4	5	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	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	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	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	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	5	6	7	8	9	A	B	B	B	B
1	0	1	2	3	4	5	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	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	5	6	6	6	6	6	6	7	8	9	A	B	C	D	E

Starting at position 1,1

For each cell:

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

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	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	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	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	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	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	5
	0	1	2	3	4	5	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	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	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	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	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	5	6	7	8	9	A	B	B	B	B
1	0	1	2	3	4	5	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	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	5	6	6	6	6	6	6	7	8	9	A	B	C	D	E

- Starting in cell with highest value (0xE), traverse matrix to the beginning

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	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	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	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	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	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	5
	0	1	2	3	4	5	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	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	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	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	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	5	6	7	8	9	A	B	B	B	B
1	0	1	2	3	4	5	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	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	5	6	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 had no alignment, and is therefore considered a variable length field
3. Followed by keyword HTTP/1.0

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.)

What Now?

- Illustrated the ability to align two sequences to each other and discern protocol 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
- Computation issues

Computation of Multiple NW

- To perform NW algorithm on multiple sequences, a hypercube would be traversed
- This leads to NP-completeness
- $2^n \times L^n$
- Where n is the number of sequences and L is the length of the sequences
- In other words, our sun will supernova before finishing the alignment 1000, 800 byte sequences

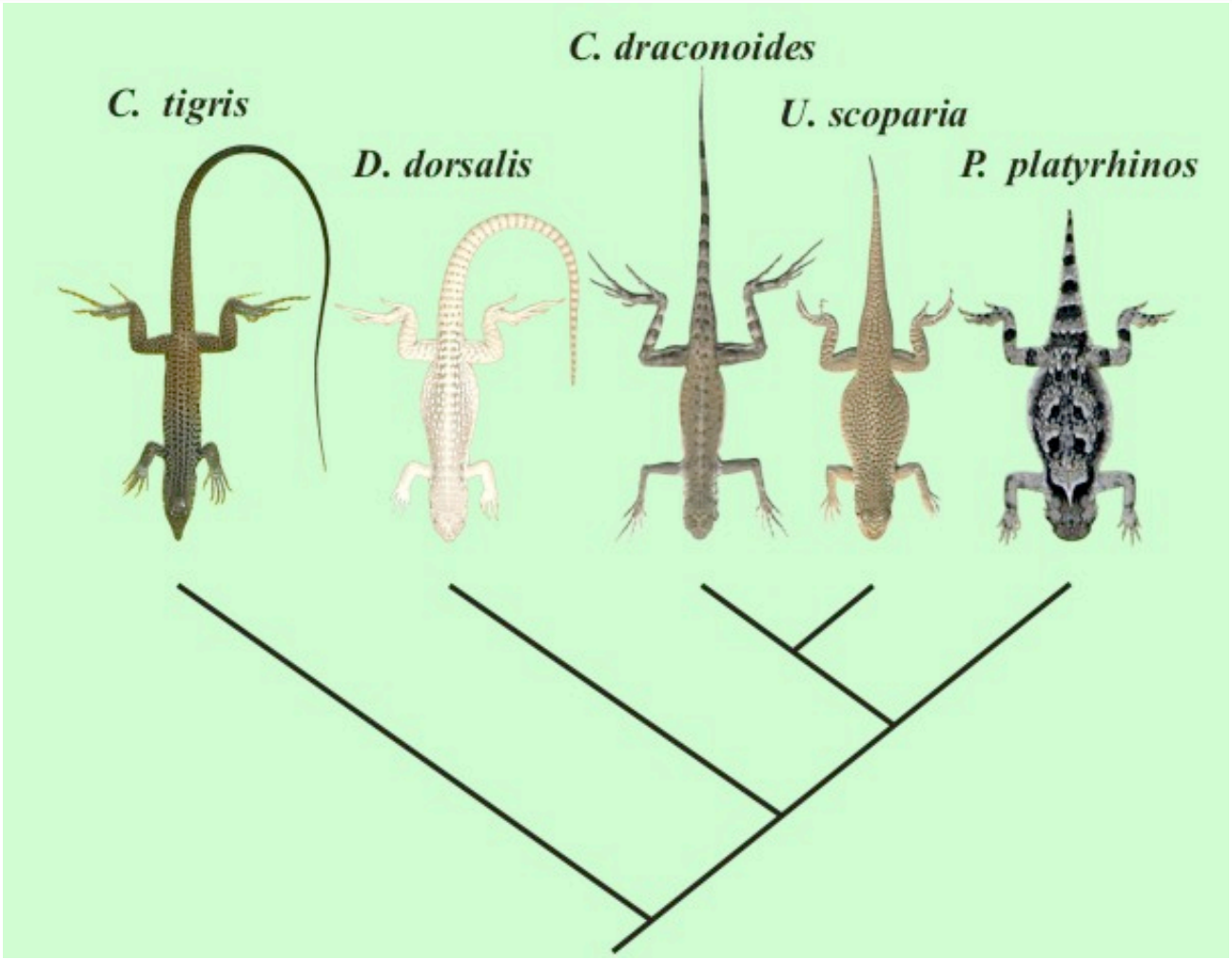
Heuristic Sequence Alignment

- 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

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

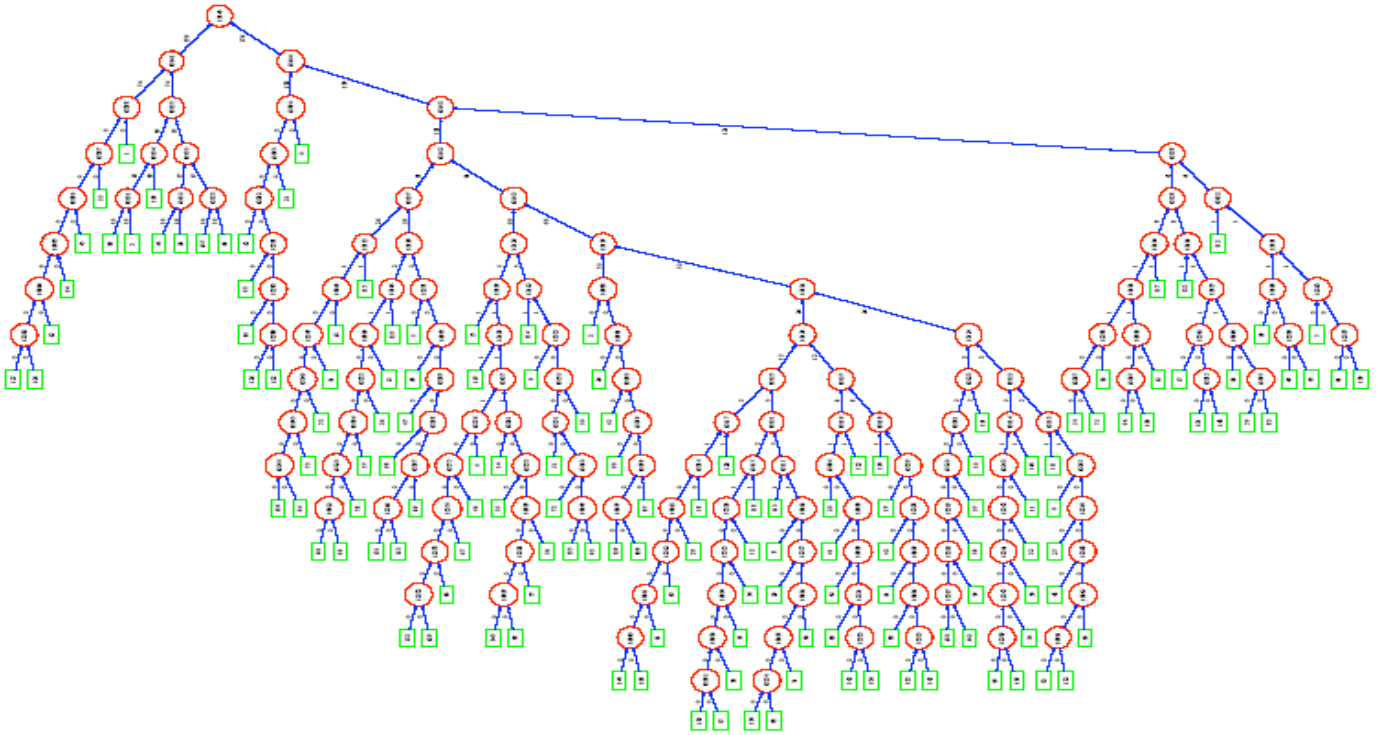
Where $d_{i,j}$ is the distance between two clusters C_i and C_j

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

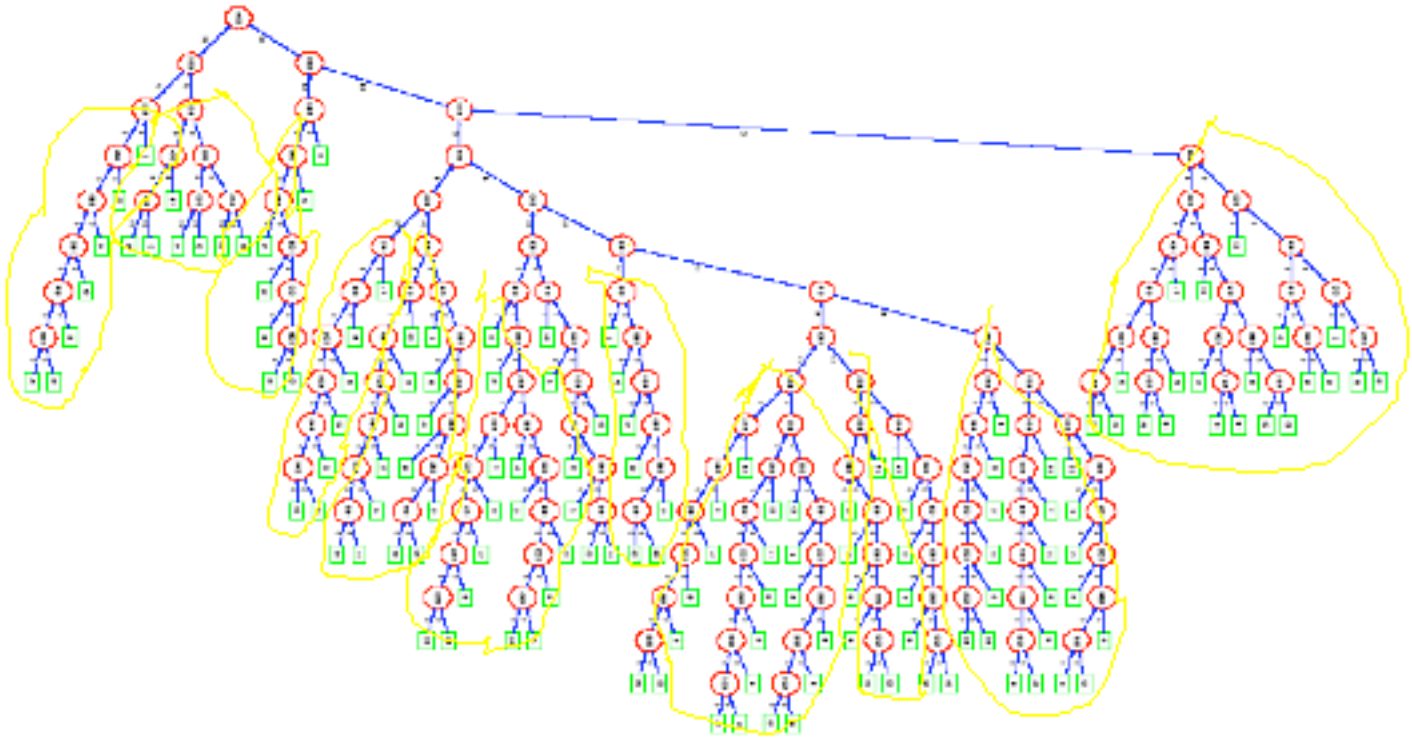
Phylogeny in Protocol Analysis

Phylogenetic tree of the SMB protocol



More Than a Pretty Picture

Phylogenetic tree of the SMB protocol



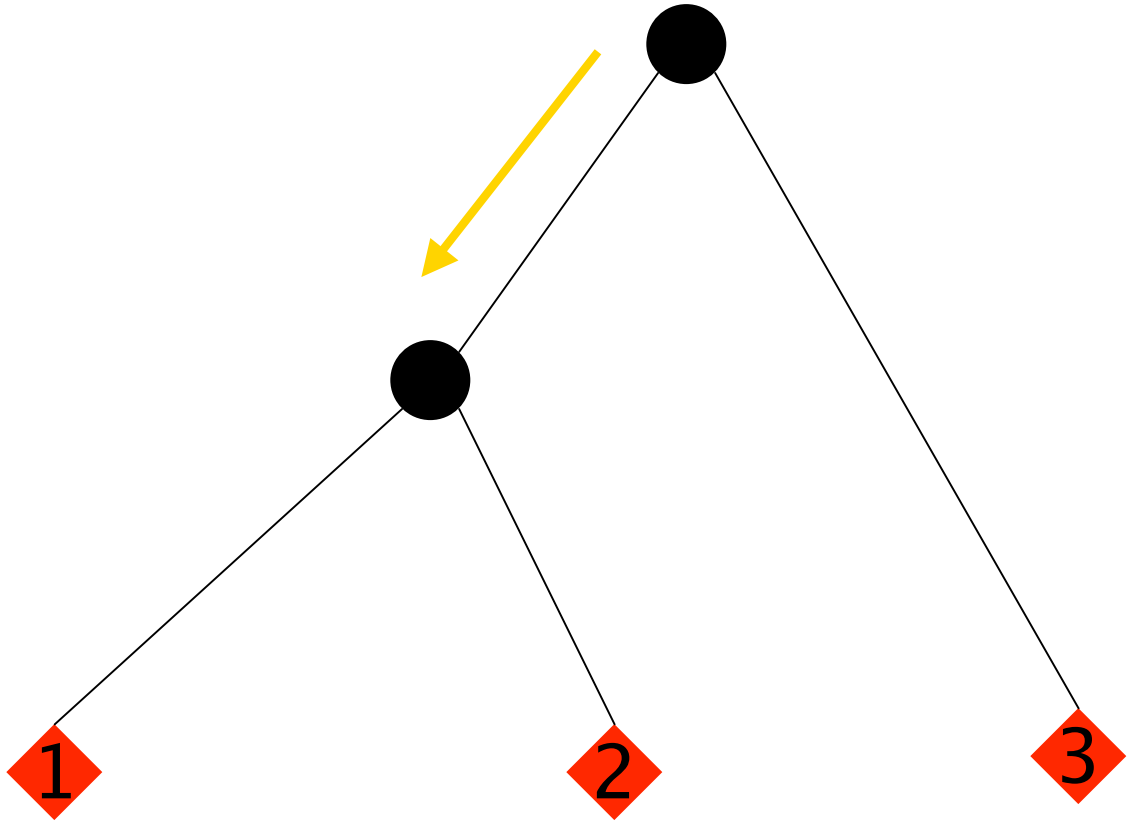
The Tree is your Guide

- Helps categorize subtypes of a particular protocol
  - SMB contains at least 11 main subtypes as illustrated
- 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.

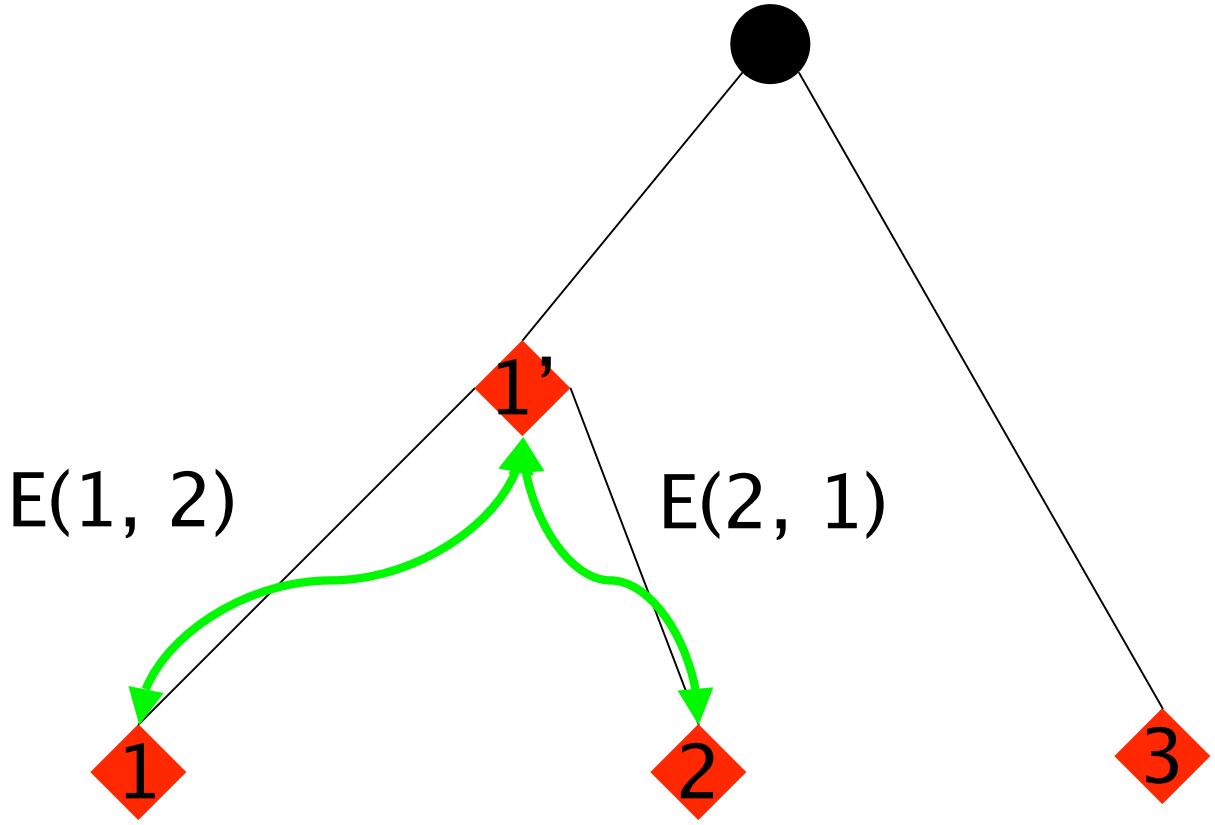
Multiple Sequence Alignment

- 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

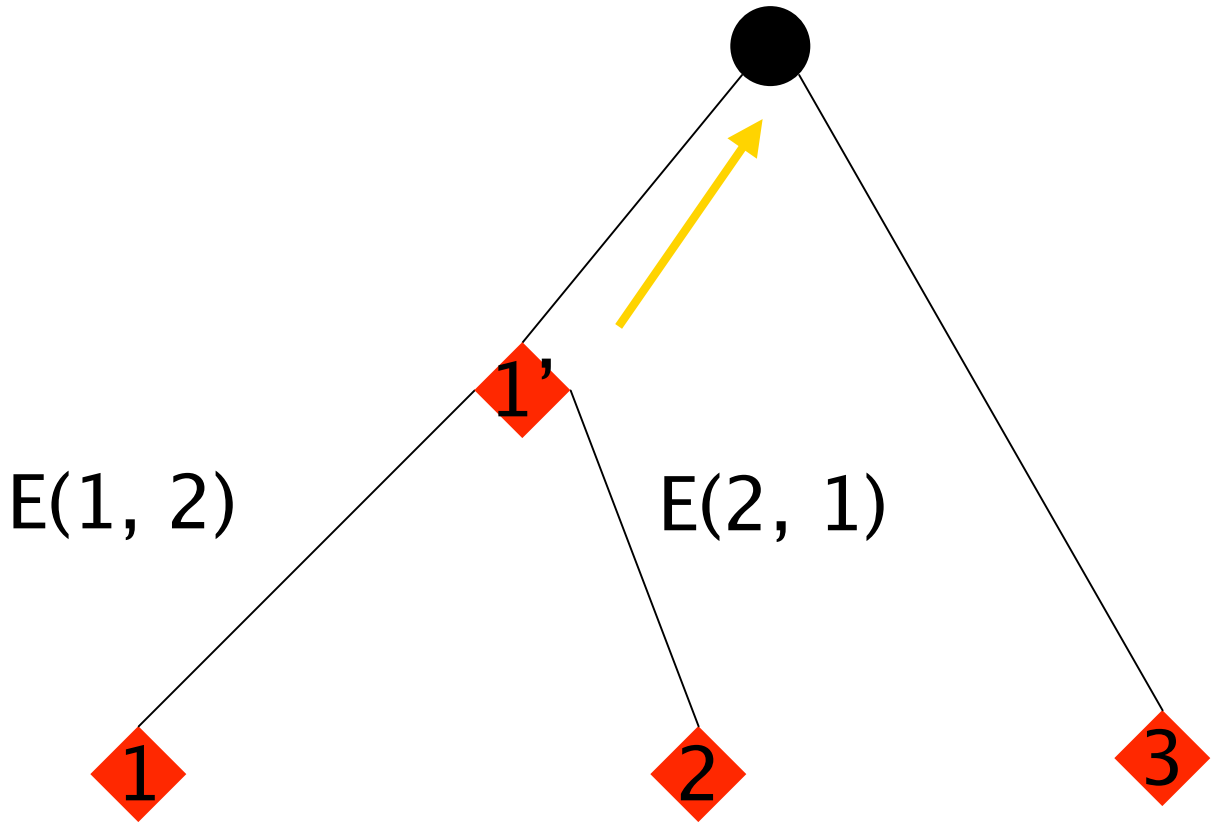
Tree Traversal Algorithm



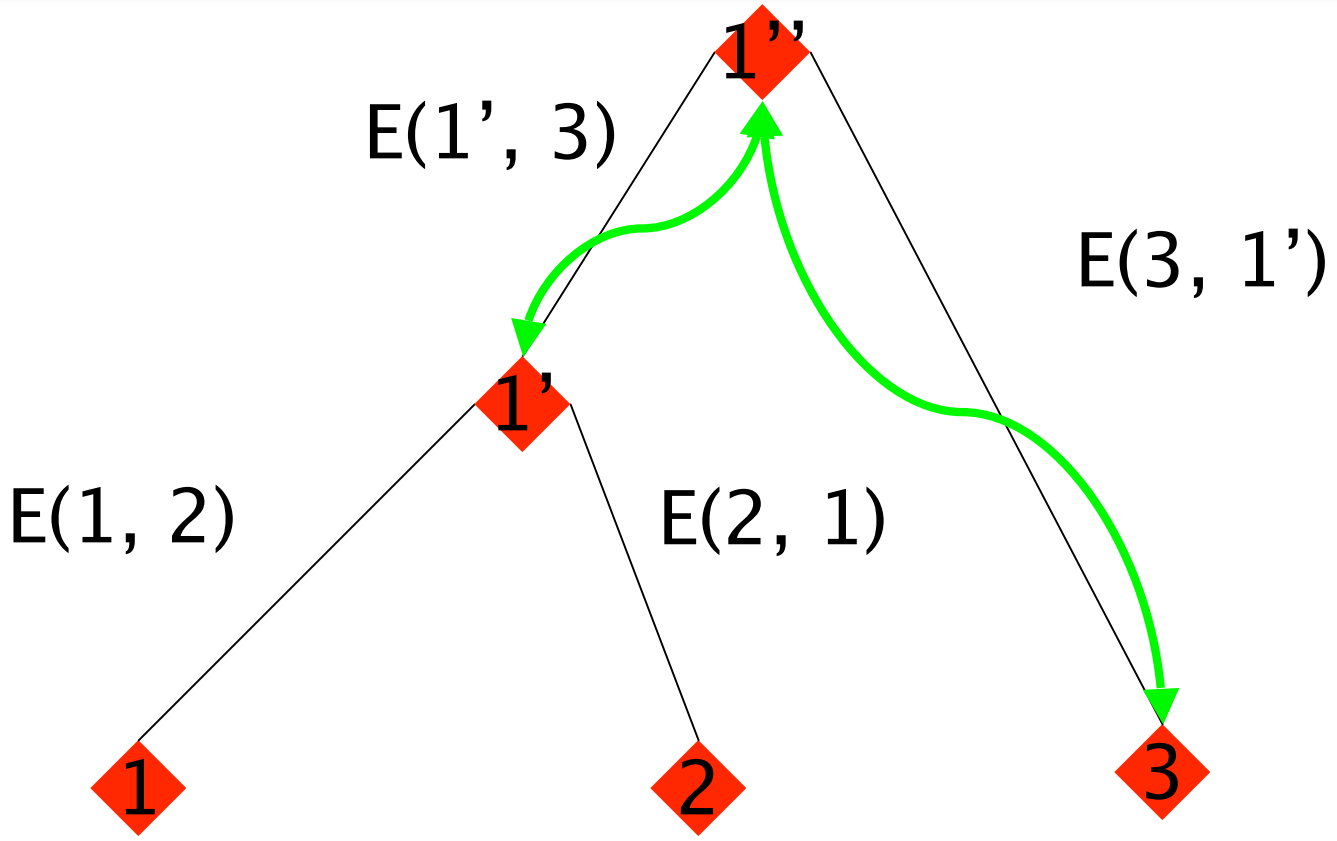
Tree Traversal Algorithm



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

Definitely works on binary protocols, but isn't as apparent on slides.

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: Sequence Ids, checksum
- 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

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 on each sequence individually, filtering out deviants
  - 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
Baseline Research http://www.baselineresearch.net/PI

If you are interested in contributing, please contact me.