Phylogenetic Comparisons of Malware

Virus Bulletin Conference 2007

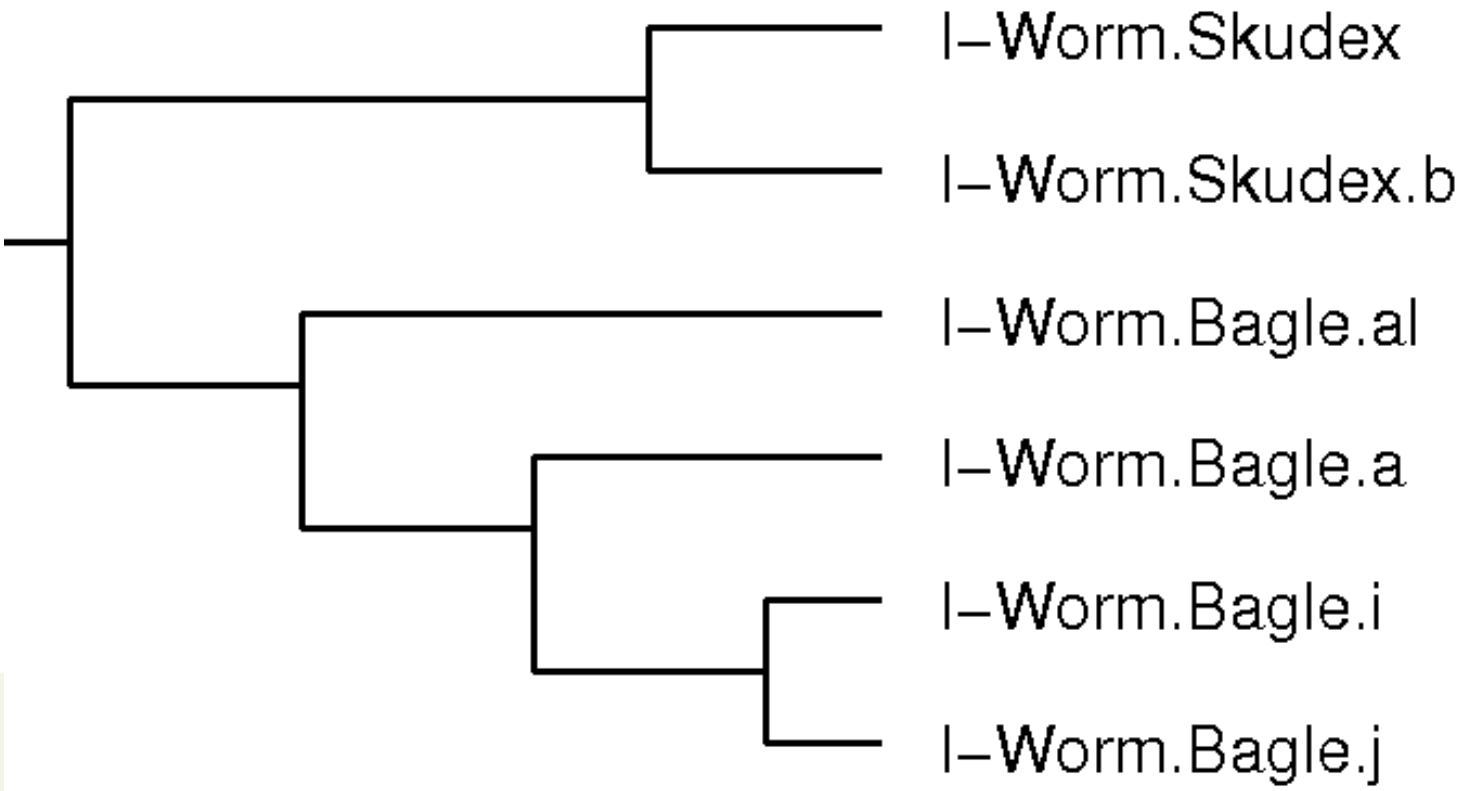
Andrew Walenstein,
Matthew Hayes, and Arun Lakhotia

Center for Advanced Computer Studies
University of Louisiana at Lafayette
walenste@ieee.org

Understanding Malware Evolution

- Long-lasting malware evolves
  - New exploits, new payloads, detection avoiding, bug fixes, etc.
  - Code is copied between families
    - Example: Bagle and Agobot
    - Both released source code: code was used elsewhere
- Q: how to understand / track evolution?
  - How to find relationships between samples?
  - How to explore found relationships?
- One approach: malware phylogenies
  - phylogeny: graph of "species" derivation relationships
  - akin to "tree of life" for biology

Example

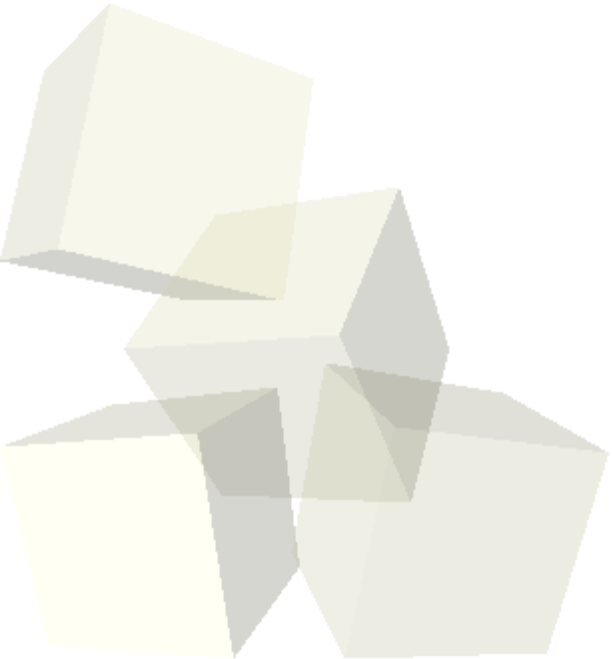

I–Worm.Skudex
I–Worm.Skudex.b
I–Worm.Bagle.al
I–Worm.Bagle.a
I–Worm.Bagle.i
I–Worm.Bagle.j

Evaluating Phylogenies

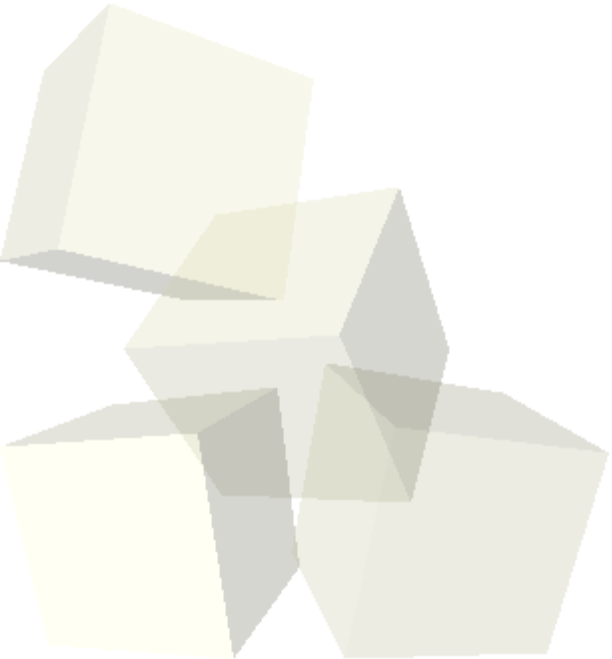
- Are phylogeny systems useful in practice?
  - little published on actually using phylogenies
    - → some pretty pictures and proof-of-concept
  - wanted a kind of case study to find out more
    - → clarification of problems and benefits in practice
    - → be able to report experiences, evaluate phylogeny extraction methods
- Target: Agobot malware families
  - Agobot source was released widely
    - → was used as basis for many different bots
    - → was available to us, enabling systematic evaluation
  - Can expected complicated evolution history
    - → easy phylogenies won't expose weaknesses

Outline of Talk

- Recap / introduce malware phylogeny methods
- Agobot study
- Summary of problems and attempted solutions



MALWARE PHYLOGENY TECHNIQUES



Origins/Parallels in Biology

- Need to reconstruct organism evolution history
  - guess relationships by examining samples
- Similarity method one of two main ways
  - Species A more similar to B than C implies A and B (probably) share a closer ancestor.
  - What is needed to computer-generate models:
    - 1. Similarity scoring function
    - 2. Graph construction algorithm based on similarity
      - common: hierarchical clustering

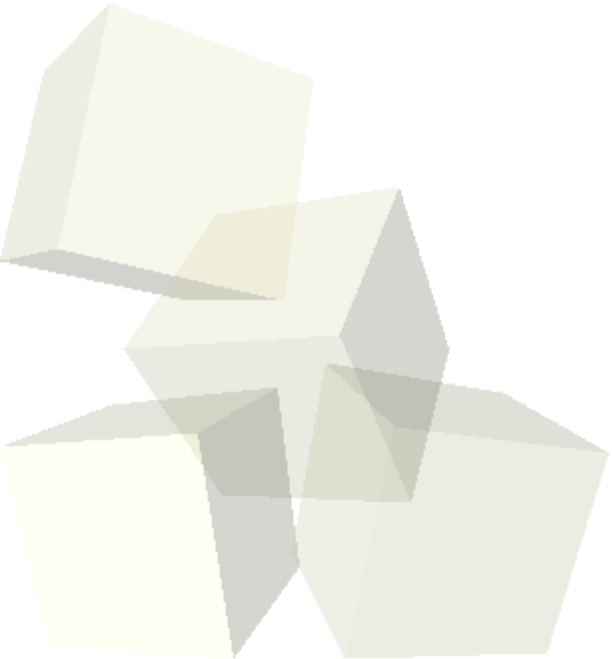
Inferring Malware Evolution

- Typical in malware phylogenies:
  - Similarity-based methods almost exclusively
  - Hierarchical clustering is typical
    - → produces strictly binary trees
    - → malware evolution known to be non-tree like
      - code sharing, for example --- a gene transfer analogue
- The similarity function often the main difference between techniques
  - different program-to-program comparisons
    - → they choose different aspects of similarity

Similarity Approaches Survey

- Control graph matching ([CE04] [DR05])
  - program similarity = flow similarity
  - (see Liang et al. in this years conference)
- Normalized Compression Distance [W05]
  - program similarity = shared information
  - idea: if to programs are similar their concatenation compresses well
- Feature vector / n-gram based [WKLP05]
  - n-gram: sequence of n characters (bytes, operations,...)
  - program similarity = feature vector similarity

APPLICATION STUDY: AGOBOT RELATED FAMILIES



Study Design

- Data sources:
  - ~4000 bot related samples
    - scanned by BitDefender: selected all “bot” related
    - unpacked & dumped using Norman Sandbox
    - 1194 distinct samples when unpacked
  - 15 bot variants constructed in vitro
    - used Agobot 3 source code
    - 15 different features turned on/off using #ifdefs
      - 2^15 different combinations possible
    - useful for producing controlled example evolution histories

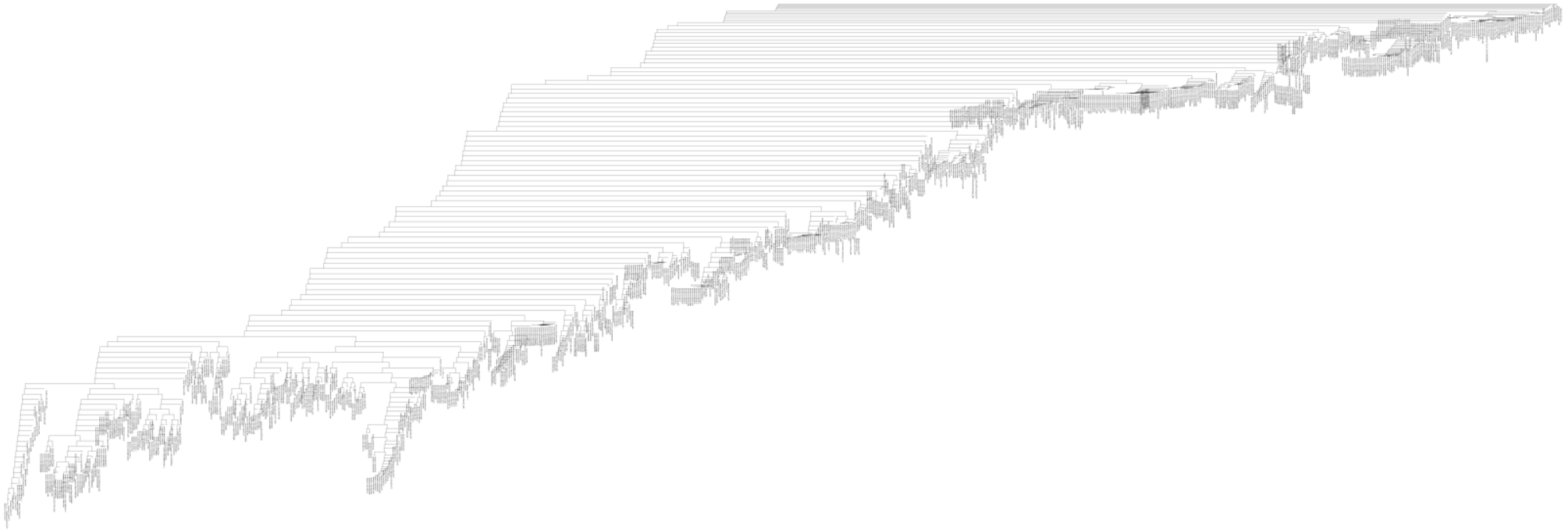
Exploratory Study

- The Plan:
  - construct phylogenies using NCD and N-gram based
  - understand main evolution features:
    (1) related families
    (2) key branch points

- The Reality:
  - NCD took several days to complete on ~1200 samples
    - (Our N-gram implementation took ~40 mins, including disassembly)

    → started with N-grams, used NCD for subsets
  - Wrestled with results, plenty of ad hoc exploration

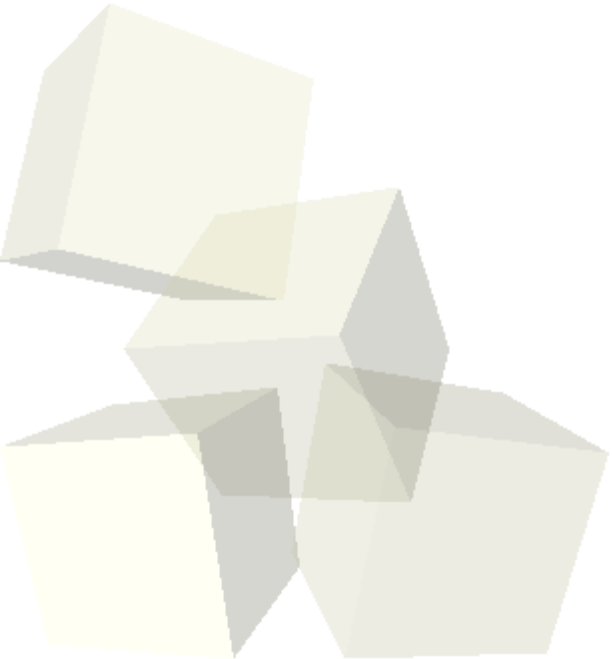
It's SO obvious



- Phylogram (tree) of all 1209 samples

Problem: Tree size

- Tree size was a significant problem
  - was not easily solved by simple zooming and panning



Dealing with Tree size

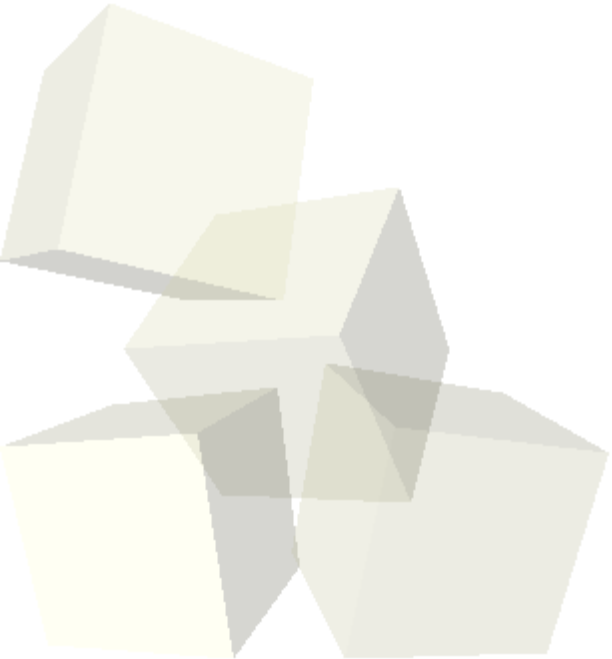
- Tried three approaches to dealing with size:
  1. Draw trees as unrooted graphs using different layout techniques, instead of "phylograms" (binary trees)
     - ➔ can help distinguish major groupings visually
  2. Merge sub-trees with high similarity & common name
     - ➔ 20 closely related SdBots in sub-tree conveys little information about overall evolution
       - family history instead of speciation events
  3. Split trees to reduce individual tree size
     - ➔ can be explored independently or compared

Splitting Trees

- Idea is to split trees at nodes where sub-trees have "low" similarity
  - for suitable definitions of "low"
  - because "low similarity" → "not useful"
    - ➔ if similarity measure working fine:
      - then samples between sub-trees are unrelated
    - ➔ if measure is just not picking up the similarity:
      - trees will be misleading in some way
      - look for other means and indicators (e.g. parallel trees)

Data from Merging & Splitting

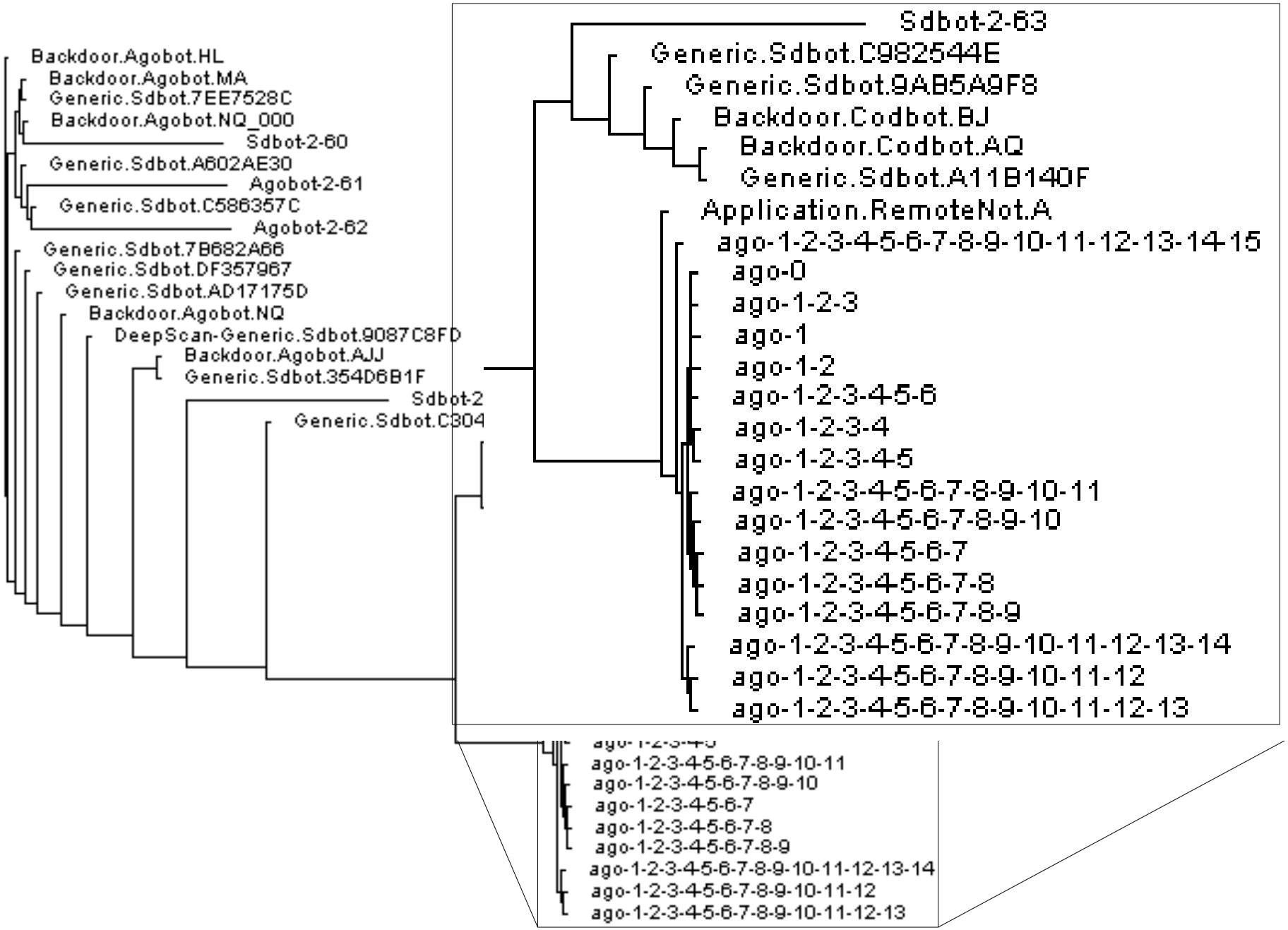
- Merging on common family names
  - 119 samples merged, < 10%
- Splitting on < .4 similarity
  - 356 splits, 308 into single leaf node trees
  - 8 trees with >10 non-leaf nodes, largest was 137 nodes



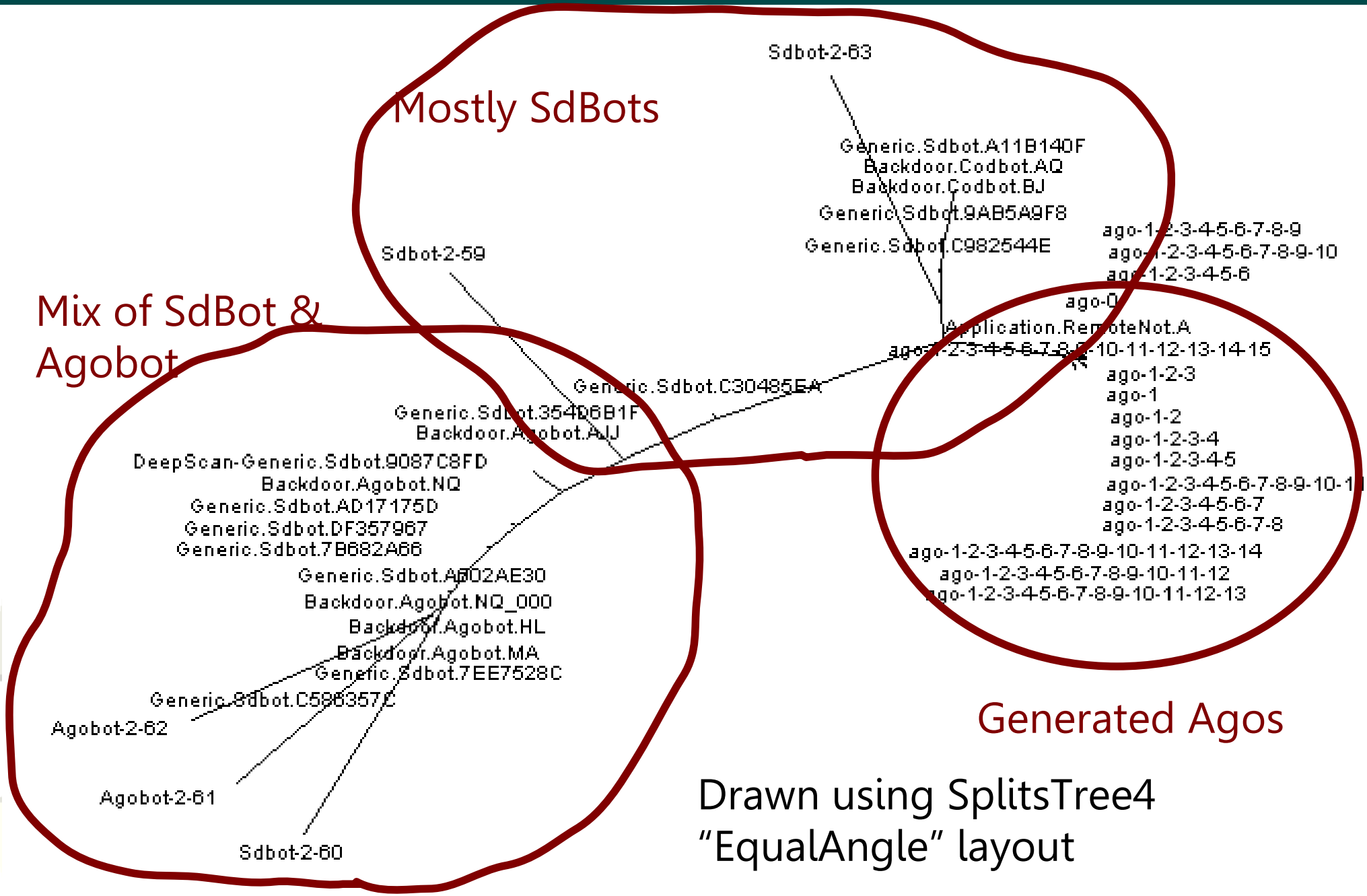
Family Characteristics

- Most trees had "mixing" of named species
  - Sdbots mixed with Rbots, IRCbots, etc.
    - → No clear separation into major lines by any technique we had available
  - data available suggested:
    - → highly interleaved development and sharing
    - → bad naming, or
    - → poor phylogenies
  - Order of 10 main branch points with multiple related variants

Example Tree (40 Nodes)

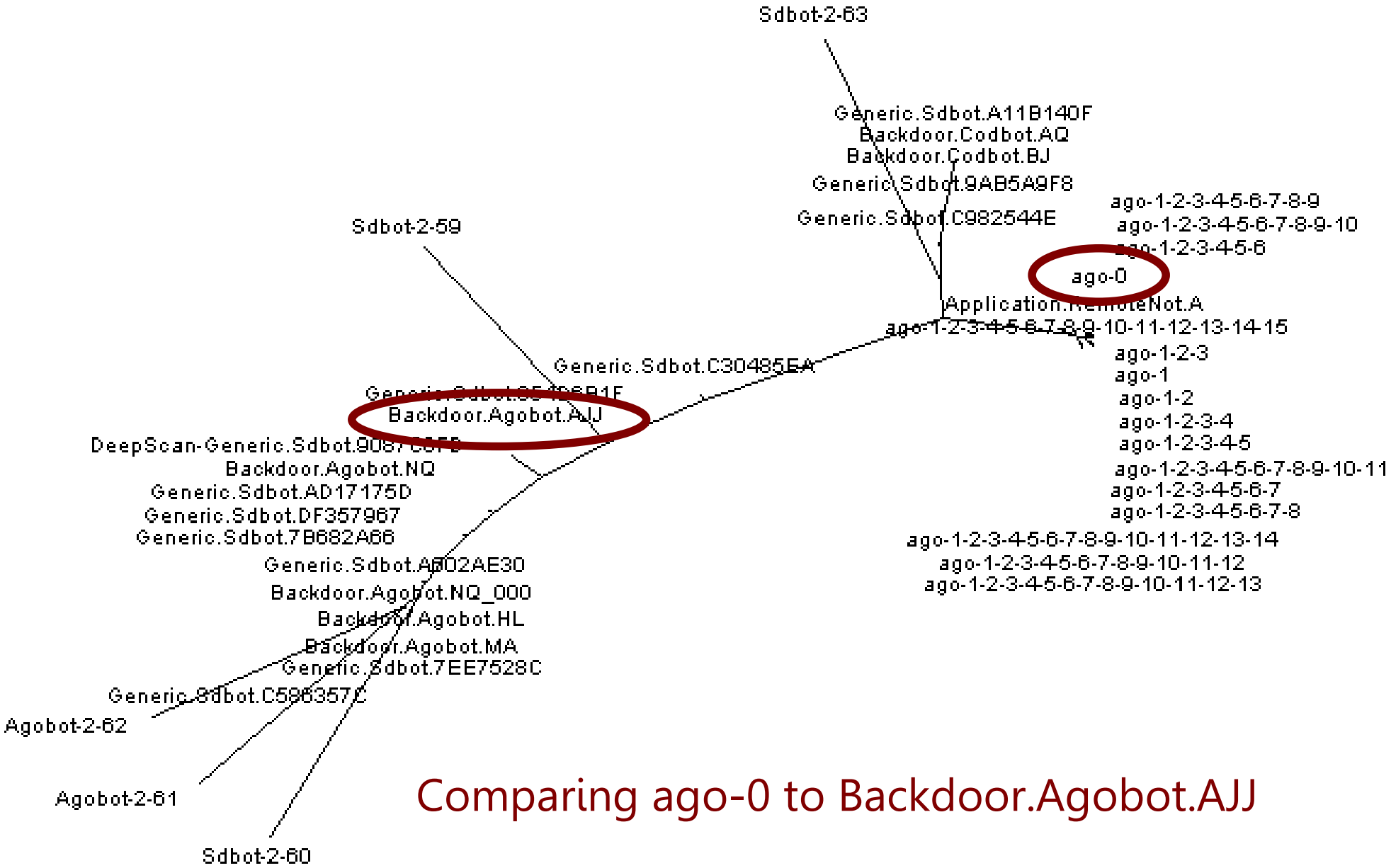


Unrooted Tree Layout



Drawn using SplitsTree4
"EqualAngle" layout

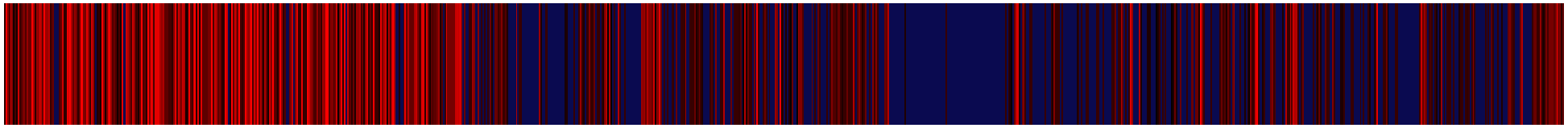
Examining Leaves



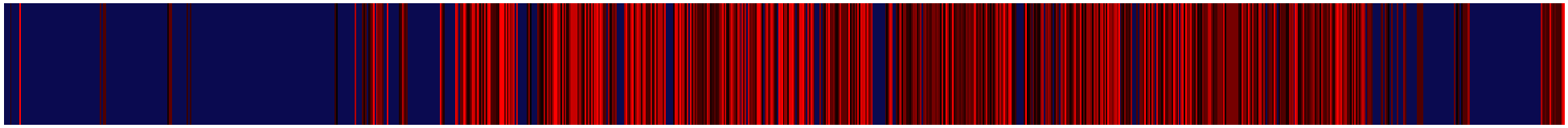
Comparing ago-0 to Backdoor.Agobot.AJJ

Examining Matches

ago-0



Backdoor.Agobot.AJJ

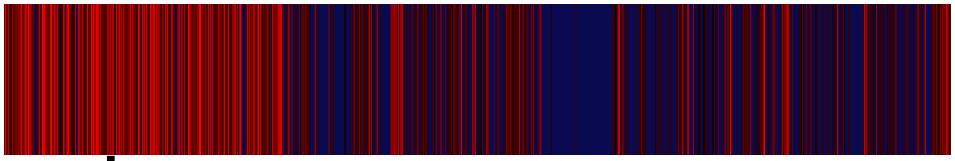


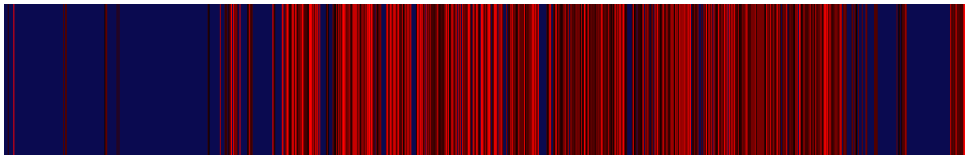
- Visualization of matches between two samples
- Legend:
  - red = match, brighter = more matching n-grams
  - blue = no match

Examining Matches

ago-0

Backdoor.Agobot.AJJ





```
bool
CDownloader::HandleCommand (CMessage * pMsg)
{ ...
  if (!pMsg->sCmd.Compare ("ftp.execute"))
    {
      if (!ParseURL (pMsg->sChatString.Token (1, " "), &uURL))
         return true;
      sUser.Assign (uURL.sUser);
...
```

- Know this from tracing source to executable

Disassembly Matching

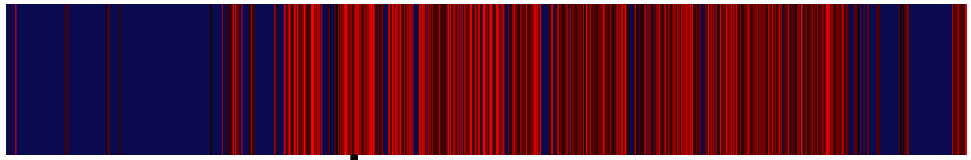
ago-0

```
mov     ecx, e
mov     [ebx+0
call    sub_40B1A1


push    offset aFtp_execute
lea     ecx, [edi+0CA8h]
call    sub_40A3D1
test    eax, eax
jnz     loc_410F68
push    esi
push    1
lea     eax, [ebp-228h]
push    eax
mov     ecx, edi
call    sub_40AC79
```

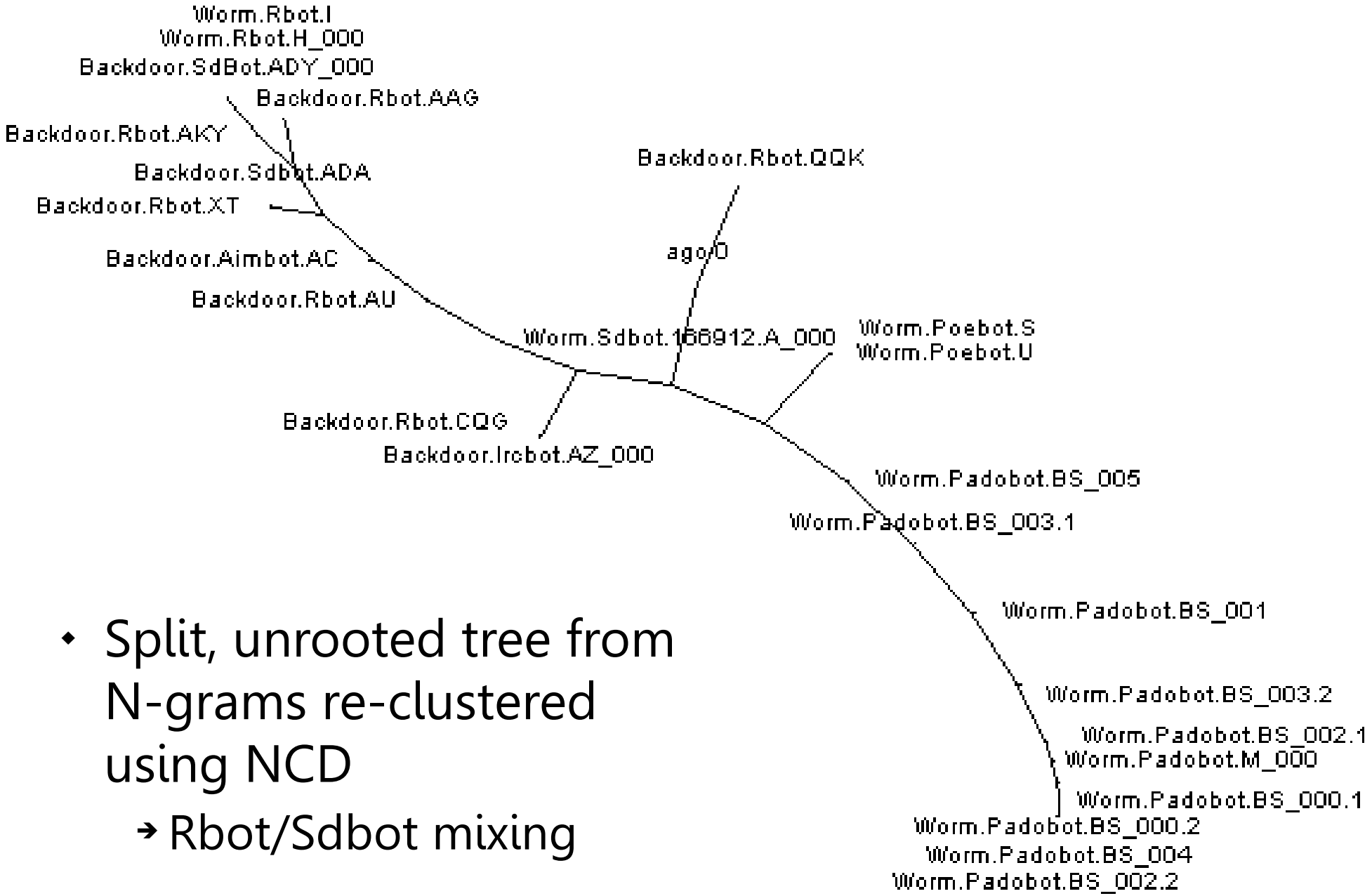
Apparent obfuscation of push immed

Backdoor.Agobot.AJJ



```
mov     ecx, ebx
mov     [ebx+6A4h], al
call    sub_58B095
mov     ebx, [ebp-10h]
lea     eax, [ebx+1818h]
push    eax
lea     ecx, [edi+670h]
call    sub_5893E1
test    eax, eax
jnz     loc_574971
push    esi
push    1
lea     eax, [ebp-23Ch]
push    eax
mov     ecx, edi
call    sub_589E18
```

Checks Using NCD



- Split, unrooted tree from N-grams re-clustered using NCD
  - → Rbot/Sdbot mixing

Discussion

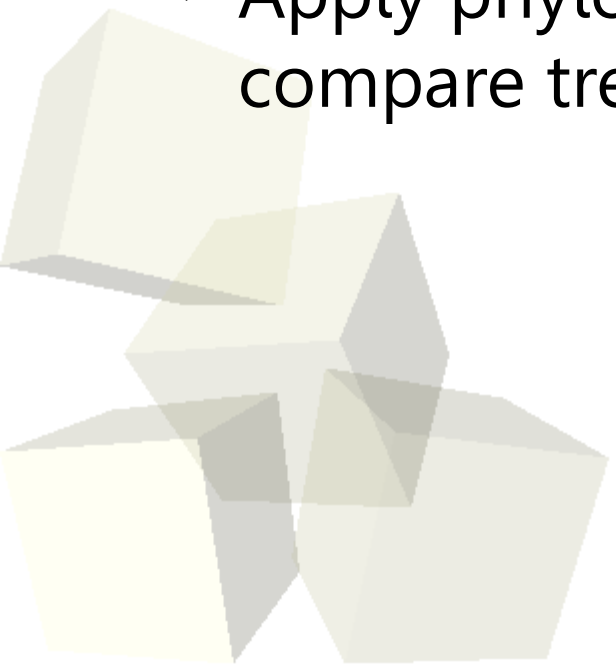
- Limitations in exploration of Agobot
  - Sample limitations:
    - ➔ collection completeness, unpacking & naming correctness
  - Phylogeny modeling limitations
    - ➔ limited selection of similarity function, clustering
- Some issues are clearer, regardless
  - Tree size and clustering issues *will remain* even if the above limitations are met
  - Question raised as to what kinds of insight will be extracted from available data and techniques
  - Tree structures may be poor choice for malware phylogenies

Conclusions / Open Questions

- Exposure of knowledge gaps / open question
  - How to provide useful analysis support?
    - → our experiences suggest a need to support:
      - splitting, merging, and alternate layouts
      - visualization, comparison, exploration
  - Need to explore network-based modeling
    - → current tree extraction may frequently be inappropriate
  - How to understand effect of data set / problem
    - → denser / better data set may help
      - wish to investigate Storm

Open Questions & Future Work

- Question of how to sytematically evaluate?
- Have been investigating controlled methods
  - Using artificial evolution trees (from Agobot and others)
    - → A priori known "correct" derivation trees
      - by construction, using automated program mutation
  - Apply phylogeny distance measures to quantitatively compare trees



Links to Online Resources

- CLUTO (glaros.dtc.umn.edu/gkhome/views/cluto)
  - feature-based and similarity-based clustering
  - output of graphs, matrices
- SRL's NCD package
  - NCD between pair files
  - generates similarity matrix in CLUTO format
  - www.cacs.louisiana.edu/labs/SRL/projects/NCD
- SplitsTree (www.splitstree.org)
  - calculates tree splits
  - multiple tree layouts

References

- [CE04] E. Carrera and G. Erdelyi
  - *Digital Genome Mapping*, Virus Bulletin 2004.
- [DR05] T. Dullien and R. Rolles
  - *Graph-based comparison of Executable Objects*, STTIC 05.
- [G05] M. Ghorghescu
  - *An Automated Virus Classification System*, Virus Bulletin, 2005.
- [HB06] D. H. Hudson and D. Bryant
  - Application of Phylogenetic Networks in Evolutionary Studies, Molecular Biology and Evolution, 23(2):254-257, 2006.
- [WKLP05] A. Walenstein, E-Md. Karim, A. Lakhotia, and L. Parida
  - *Malware Phylogeny Generation Using Permutations of Code*, Journal in Computer Virology, v1.1, 2005.
- [W05] S. Wehner
  - *Analyzing Worms and Network Traffic using Compression,* arXiv:cs.CR/0504045 v1 12 Apr 2005

THANK YOU

- Thanks to
  - Michael Venable, Software Engineer
  - Mohamed Chouchane, Ph.D. Student
  - & the whole Software Research Laboratory (SRL) team!
- Funding by
  - Louisiana IT Initiative