Networks for Exploring the Evolution of Pandemic H1N1 Influenza and Covid-19

Shahid Bokhari

Algopath LLC

shb@algopath.com

CLSAC 2020

Overview

Overview
Viruses
Evolution
Covid-19
Phylogenetic Trees
Tree/web of Life
Antigenic Shift
Results
Model
Layered Graph
Features
Example Code
Performance
Workflow
Tree
Bottleneck Viruses
Bottleneck Viruses
Extensions
Future

- Viruses
- Reassortment/Recombination Networks
  - Influenza A
  - Reassortment in segmented viruses
  - Covid-19
  - Recombination in Coronaviruses
  - Limitations of Phylogenetic Trees
  - Networks instead of Trees
  - Implementation and Results for Influenza
- Projected analysis for Covid-19

Viruses: A threat to Civilization

- Global Health Significance

Viruses: A threat to Civilization

- Global Health Significance
- Different variants occur from time to time

Overview
Viruses
Evolution
Covid-19
Phylogenetic Trees
Tree/web of Life
Antigenic Shift
Results
Model
Layered Graph
Features
Example Code
Performance
Workflow
Tree
Bottleneck Viruses
Bottleneck Viruses
Extensions
Future

Viruses: A threat to Civilization

- Global Health Significance
- Different variants occur from time to time
- ~~Fear~~ Reality of Global Pandemic
    - 1918 “Spanish” flu
    - 1976 panic
    - 2009 H1N1 panic
    - Current Covid-19
    - Immense socio-economic significance

Viruses: A threat to Civilization

- Global Health Significance
- Different variants occur from time to time
- ~~Fear~~ Reality of Global Pandemic
  - 1918 “Spanish” flu
  - 1976 panic
  - 2009 H1N1 panic
  - Current Covid-19
  - Immense socio-economic significance
- Segmented Viruses
  - Influenza, Rotaviruses, Congo fever, Rift Valley fever

Viruses: A threat to Civilization

- Global Health Significance
- Different variants occur from time to time
- ~~Fear~~ Reality of Global Pandemic
  - 1918 “Spanish” flu
  - 1976 panic
  - 2009 H1N1 panic
  - Current Covid-19
  - Immense socio-economic significance
- Segmented Viruses
  - Influenza, Rotaviruses, Congo fever, Rift Valley fever
- Unsegmented Viruses
  - Coronavirus, MERS, SARS

Structure & Evolution of Influenza-A

- Genome: 8 RNA segments
  (900–2,400 bases each, total ≈14,000)
- Evolves by:
  - Mutation *(antigenic drift)*
  - Reassortment *(antigenic shift)*
    - Entire segments interchanged
    - Happens when two strains coinfect a single cell
    - ⇒ large jumps in composition of genome
    - ⇒ strains that are especially contagious
    - Likely occurs when humans, birds and swine live in close proximity

Covid-19

- Genome: 29,903 bases RNA non-segmented (about 25,000 total genomes)
- Evolves by:
  - Mutation
  - Recombination
    - Portions of genome interchanged
    - Happens when two strains coinfect a single cell
    - ⇒ large jumps in composition of genome
    - ⇒ strains that are especially contagious
    - Perhaps occurred when bats and pangolins were in close proximity (Wuhan market?)–no agreement about this.
    - However, these animals would not encounter each other in natural circumstances.

Phylogenetic Trees

- Classical tools for analyzing evolution
- Assume everything originated from one source
- Cannot capture reassortment, recombination or other forms of *Horizontal Gene Transfer (HGT)*
- More general structures–Networks–needed

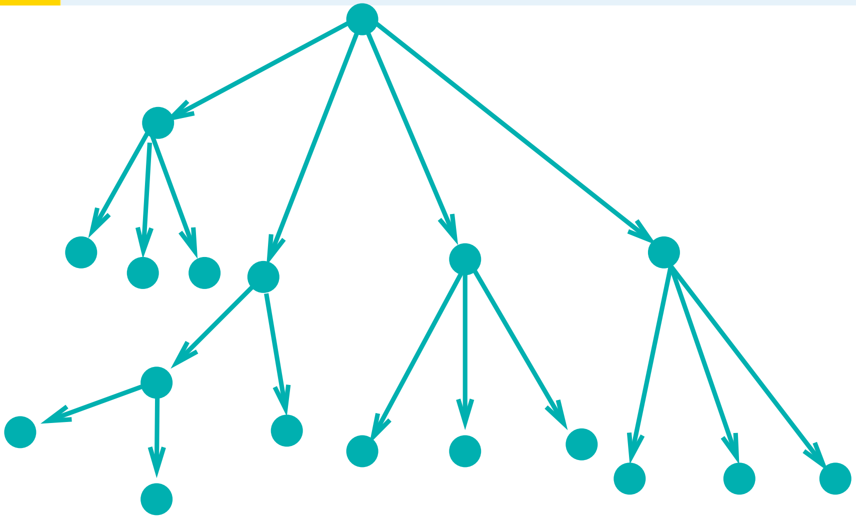
Phylogenetic Trees

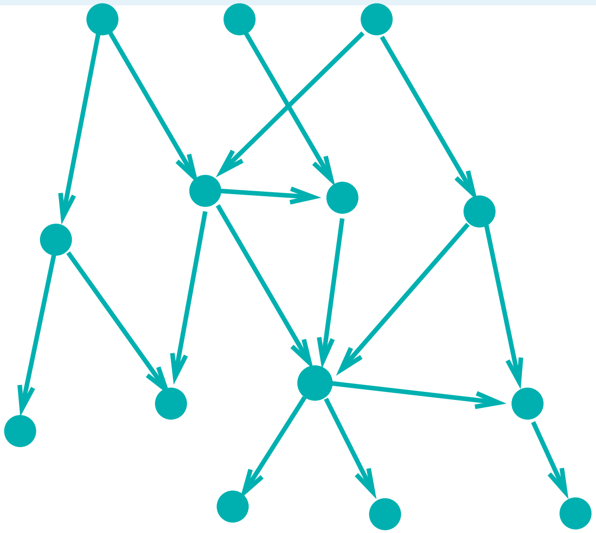
- Classical tools for analyzing evolution
- Assume everything originated from one source
- Cannot capture reassortment, recombination or other forms of *Horizontal Gene Transfer (HGT)*
- More general structures–Networks–needed
- Phylogenetic trees completely control our thinking about evolution

Phylogenetic Trees

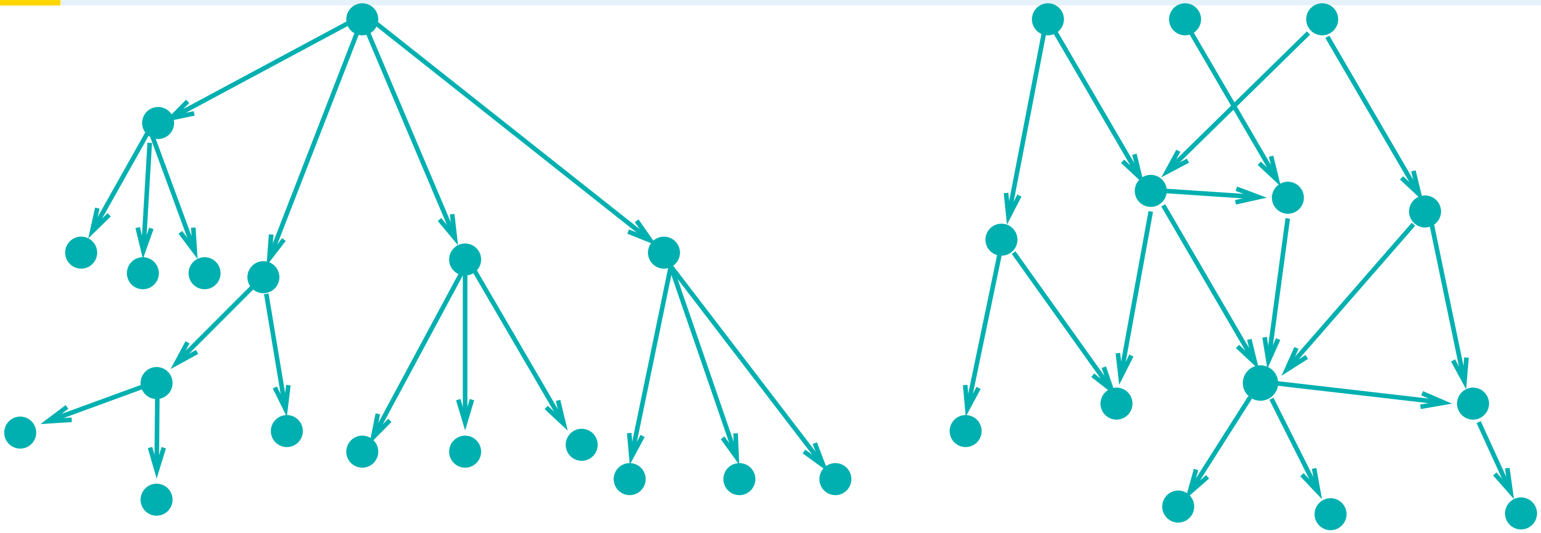
- Classical tools for analyzing evolution
- Assume everything originated from one source
- Cannot capture reassortment, recombination or other forms of *Horizontal Gene Transfer (HGT)*
- More general structures–Networks–needed
- Phylogenetic trees completely control our thinking about evolution
- However they are inadequate to fully explain evolution

The ~~Tree~~ Web of Life





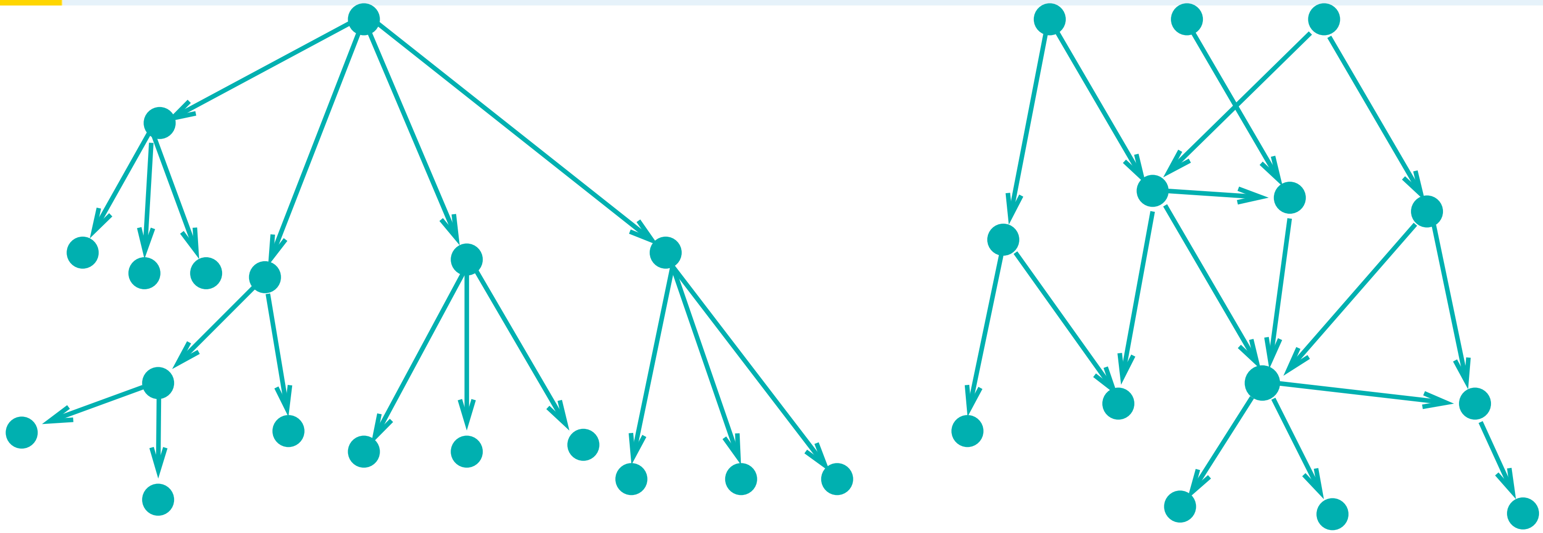
The ~~Tree~~ Web of Life



Molecular phylogeneticists will [fail] to find the “true tree,” not because their methods are inadequate. . . but because the history of life cannot be represented as a tree.

W. F. Doolittle, Science, 2124-2128, vol. 284, 25 June 1999.

The ~~Tree~~ Web of Life

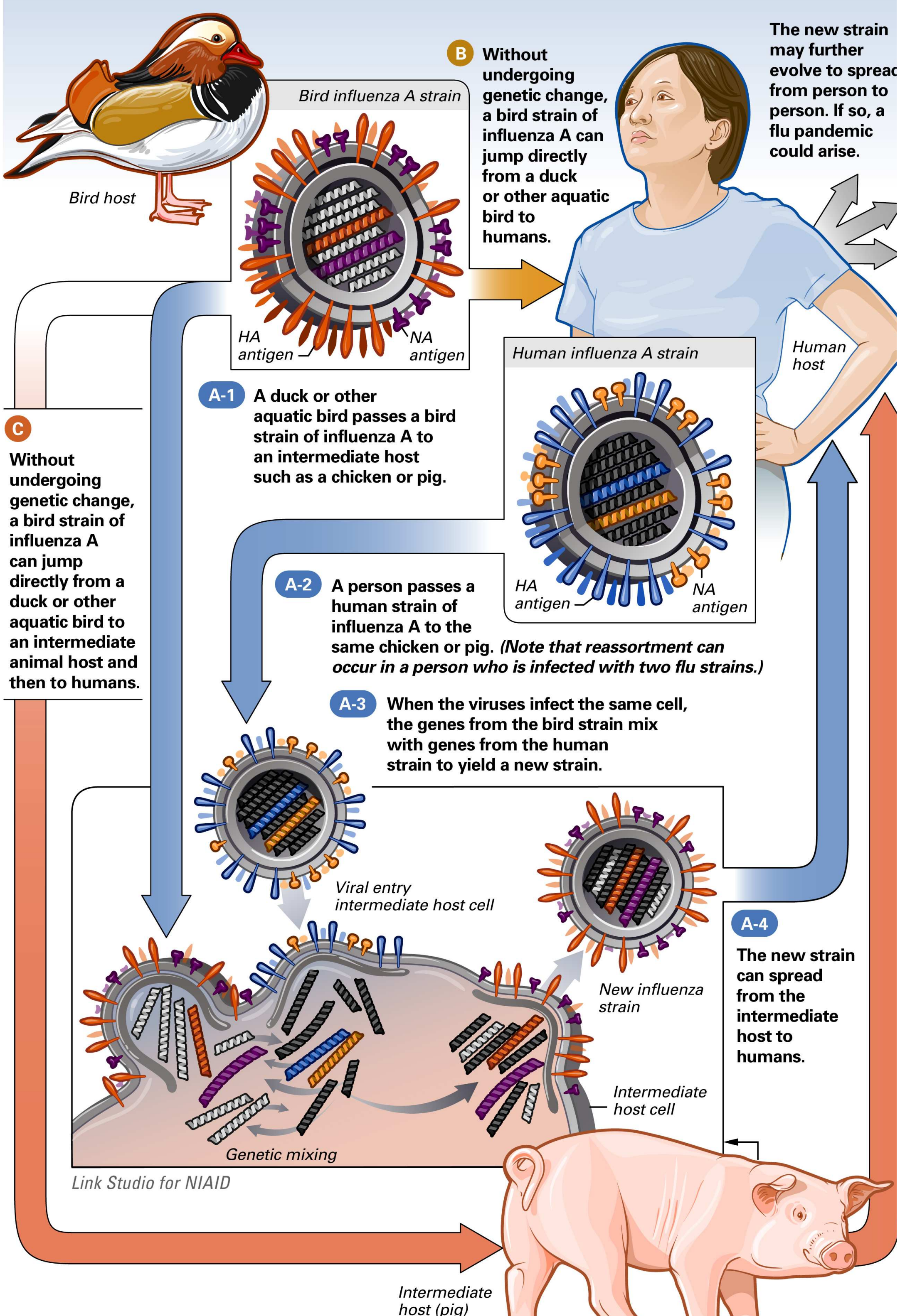


. . . the universal phylogenetic tree. . . is no more than a graphic device. . . it is not a matter of whether your data are consistent with a tree, but whether tree topology is a useful way to represent your data. . . Under conditions of extreme [Horizontal Gene Transfer], there is no (organismal) “tree.” Evolution is basically reticulate (*network-like*).

C. R. Woese, Microbiology and Molec. Biol. Rev., June 2004, pp 173–186.

Reassortment in segmented viruses–the case of influenza

(niaid.nih.gov)



B Without undergoing genetic change, a bird strain of influenza A can jump directly from a duck or other aquatic bird to humans.

The new strain may further evolve to spread from person to person. If so, a flu pandemic could arise.

C Without undergoing genetic change, a bird strain of influenza A can jump directly from a duck or other aquatic bird to an intermediate animal host and then to humans.

A-1 A duck or other aquatic bird passes a bird strain of influenza A to an intermediate host such as a chicken or pig.

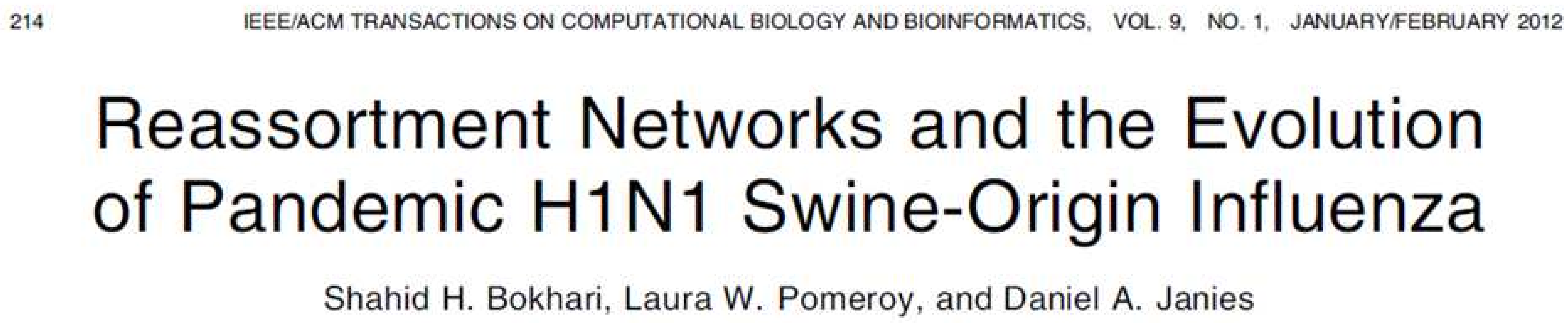
A-2 A person passes a human strain of influenza A to the same chicken or pig. *(Note that reassortment can occur in a person who is infected with two flu strains.)*

A-3 When the viruses infect the same cell, the genes from the bird strain mix with genes from the human strain to yield a new strain.

A-4 The new strain can spread from the intermediate host to humans.

Link Studio for NIAID

Results



214 IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 9, NO. 1, JANUARY/FEBRUARY 2012

Reassortment Networks and the Evolution of Pandemic H1N1 Swine-Origin Influenza

Shahid H. Bokhari, Laura W. Pomeroy, and Daniel A. Janies

Research supported by Pacific Northwest National Laboratory, John Feo, Director of the division maintaining the (late) Cray XMT supercomputer.

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses $=$ nodes

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses $=$ nodes
- Reassortment events $=$ nodes

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses = nodes
- Reassortment events = nodes
- Mutations/reassortment choices = edges

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses $=$ nodes
- Reassortment events $=$ nodes
- Mutations/reassortment choices $=$ edges
- Weights on edges represent edit distances

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses = nodes
- Reassortment events = nodes
- Mutations/reassortment choices = edges
- Weights on edges represent edit distances
- We assume that some *observed* viruses are the ancestors to (or have reassorted with) other *observed* viruses.

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses = nodes
- Reassortment events = nodes
- Mutations/reassortment choices = edges
- Weights on edges represent edit distances
- We assume that some *observed* viruses are the ancestors to (or have reassorted with) other *observed* viruses.
- Unlike typical phylogenetics, which uses observed *taxa* and builds tree by *inferring* ancestors.
  (Taxon = observed organism at leaf of tree.)

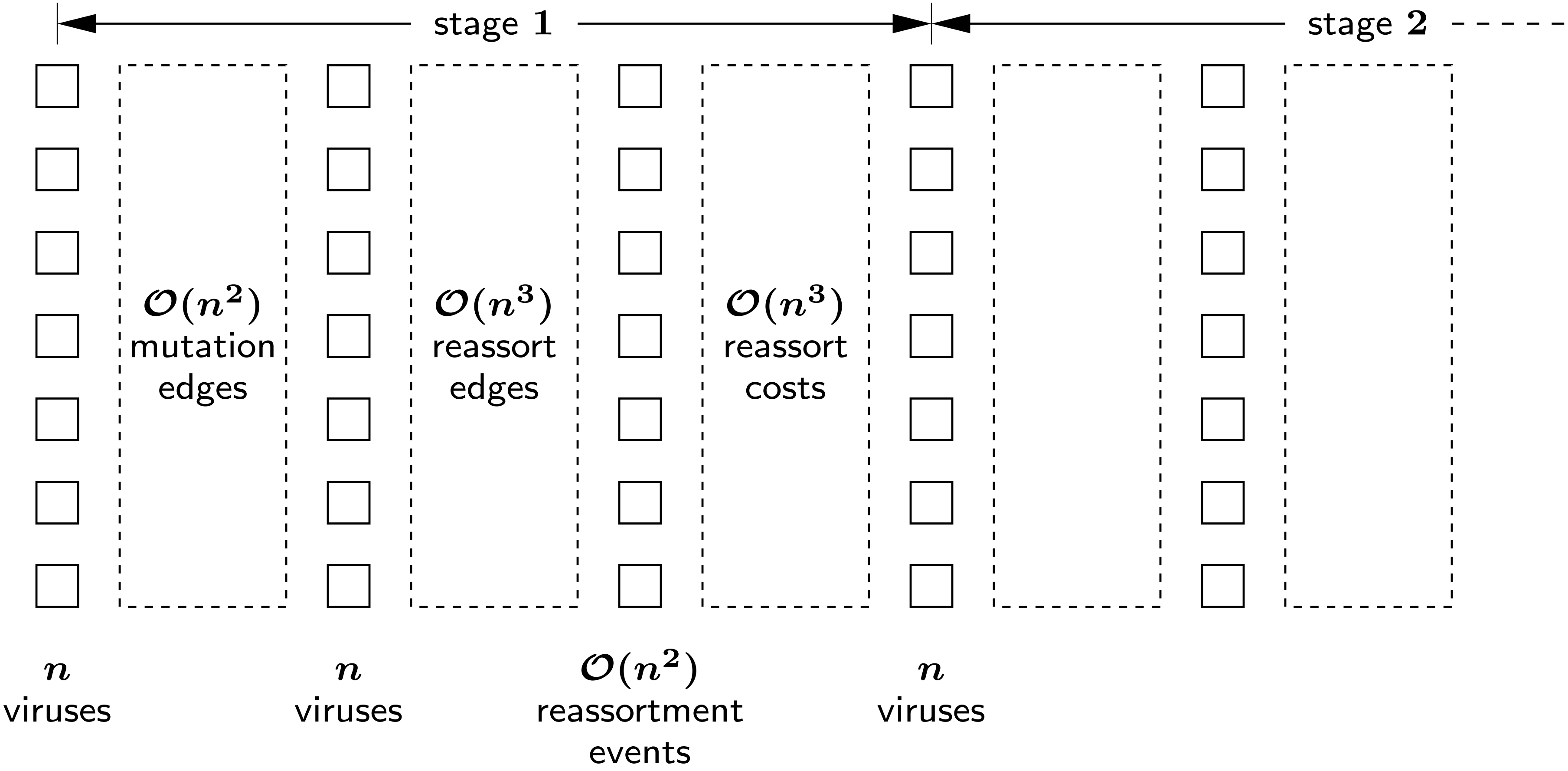
Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses = nodes
- Reassortment events = nodes
- Mutations/reassortment choices = edges
- Weights on edges represent edit distances
- We assume that some *observed* viruses are the ancestors to (or have reassorted with) other *observed* viruses.
- Unlike typical phylogenetics, which uses observed *taxa* and builds tree by *inferring* ancestors.
  (Taxon = observed organism at leaf of tree.)
- Path = sequence of mutations, reassortments and stasis

Graph Theoretic Model

- Evolution over τ stages or seasons modeled with a layered or *multipartite* graph.
- Viruses = nodes
- Reassortment events = nodes
- Mutations/reassortment choices = edges
- Weights on edges represent edit distances
- We assume that some *observed* viruses are the ancestors to (or have reassorted with) other *observed* viruses.
- Unlike typical phylogenetics, which uses observed *taxa* and builds tree by *inferring* ancestors.
  (Taxon = observed organism at leaf of tree.)
- Path = sequence of mutations, reassortments and stasis
- Path length = cost of evolution: lower cost $\Rightarrow$ smaller sum of evolutionary distances between successive viruses.

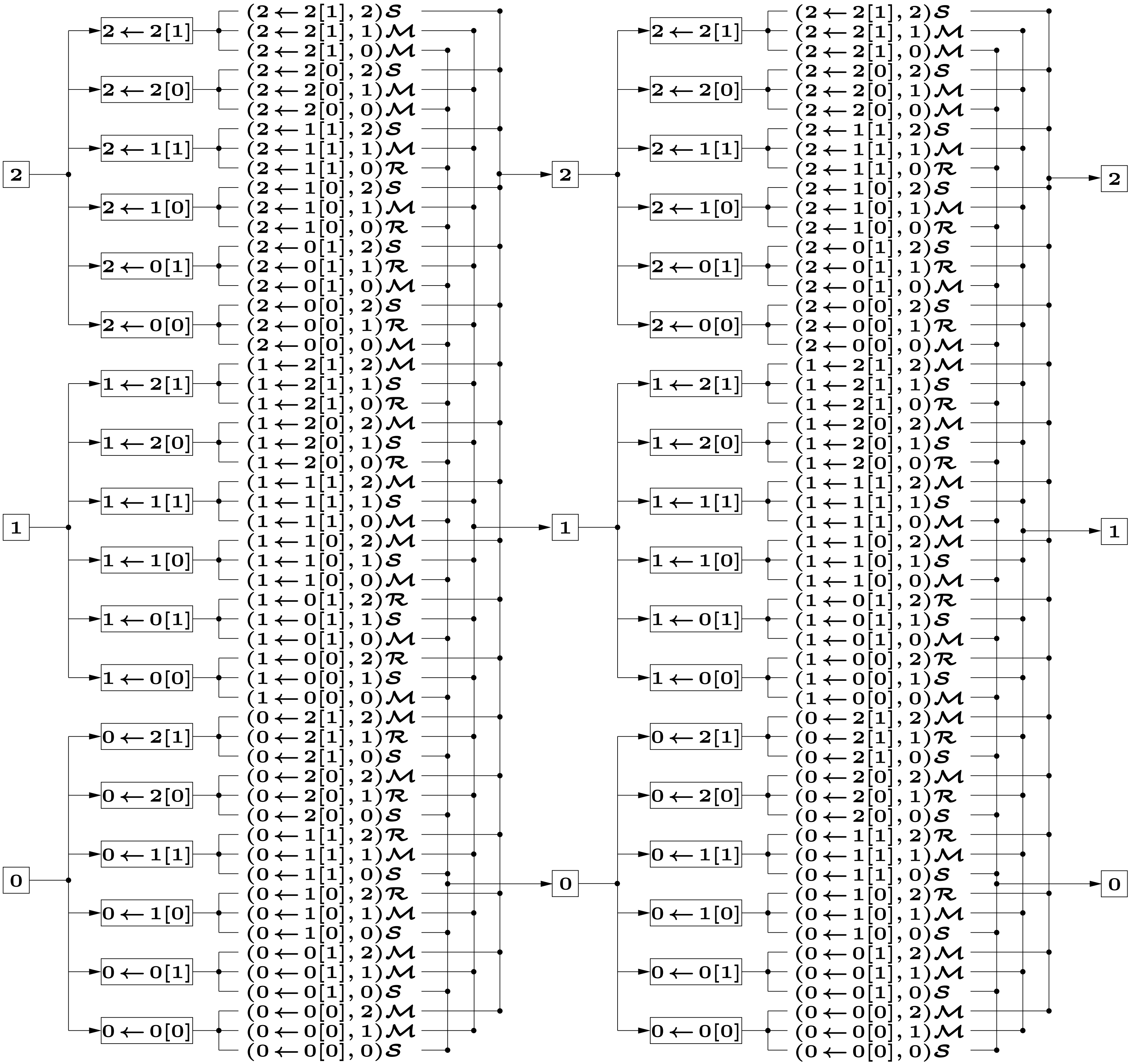
Layered Graph


stage 1
stage 2
$\mathcal{O}(n^2)$ mutation edges
$\mathcal{O}(n^3)$ reassort edges
$\mathcal{O}(n^3)$ reassort costs
n viruses
n viruses
$\mathcal{O}(n^2)$ reassortment events
n viruses

However, an interesting result: Separate mutation and reassortment edges not needed.

Mutation and Stasis are special cases of Reassortment *from our graph-theoretic viewpoint*

Network for
n =3 viruses,
τ =2 stages,
2 segments/virus



2
1
0
$2 \leftarrow 2[1]$
$(2 \leftarrow 2[1], 2)\mathcal{S}$
$(2 \leftarrow 2[1], 1)\mathcal{M}$
$(2 \leftarrow 2[1], 0)\mathcal{M}$
$2 \leftarrow 2[0]$
$(2 \leftarrow 2[0], 2)\mathcal{S}$
$(2 \leftarrow 2[0], 1)\mathcal{M}$
$(2 \leftarrow 2[0], 0)\mathcal{M}$
$2 \leftarrow 1[1]$
$(2 \leftarrow 1[1], 2)\mathcal{S}$
$(2 \leftarrow 1[1], 1)\mathcal{M}$
$(2 \leftarrow 1[1], 0)\mathcal{R}$
$2 \leftarrow 1[0]$
$(2 \leftarrow 1[0], 2)\mathcal{S}$
$(2 \leftarrow 1[0], 1)\mathcal{M}$
$(2 \leftarrow 1[0], 0)\mathcal{R}$
$2 \leftarrow 0[1]$
$(2 \leftarrow 0[1], 2)\mathcal{S}$
$(2 \leftarrow 0[1], 1)\mathcal{R}$
$(2 \leftarrow 0[1], 0)\mathcal{M}$
$2 \leftarrow 0[0]$
$(2 \leftarrow 0[0], 2)\mathcal{S}$
$(2 \leftarrow 0[0], 1)\mathcal{R}$
$(2 \leftarrow 0[0], 0)\mathcal{M}$
$1 \leftarrow 2[1]$
$(1 \leftarrow 2[1], 2)\mathcal{M}$
$(1 \leftarrow 2[1], 1)\mathcal{S}$
$(1 \leftarrow 2[1], 0)\mathcal{R}$
$1 \leftarrow 2[0]$
$(1 \leftarrow 2[0], 2)\mathcal{M}$
$(1 \leftarrow 2[0], 1)\mathcal{S}$
$(1 \leftarrow 2[0], 0)\mathcal{R}$
$1 \leftarrow 1[1]$
$(1 \leftarrow 1[1], 2)\mathcal{M}$
$(1 \leftarrow 1[1], 1)\mathcal{S}$
$(1 \leftarrow 1[1], 0)\mathcal{M}$
$1 \leftarrow 1[0]$
$(1 \leftarrow 1[0], 2)\mathcal{M}$
$(1 \leftarrow 1[0], 1)\mathcal{S}$
$(1 \leftarrow 1[0], 0)\mathcal{M}$
$1 \leftarrow 0[1]$
$(1 \leftarrow 0[1], 2)\mathcal{R}$
$(1 \leftarrow 0[1], 1)\mathcal{S}$
$(1 \leftarrow 0[1], 0)\mathcal{M}$
$1 \leftarrow 0[0]$
$(1 \leftarrow 0[0], 2)\mathcal{R}$
$(1 \leftarrow 0[0], 1)\mathcal{S}$
$(1 \leftarrow 0[0], 0)\mathcal{M}$
$0 \leftarrow 2[1]$
$(0 \leftarrow 2[1], 2)\mathcal{M}$
$(0 \leftarrow 2[1], 1)\mathcal{R}$
$(0 \leftarrow 2[1], 0)\mathcal{S}$
$0 \leftarrow 2[0]$
$(0 \leftarrow 2[0], 2)\mathcal{M}$
$(0 \leftarrow 2[0], 1)\mathcal{R}$
$(0 \leftarrow 2[0], 0)\mathcal{S}$
$0 \leftarrow 1[1]$
$(0 \leftarrow 1[1], 2)\mathcal{R}$
$(0 \leftarrow 1[1], 1)\mathcal{M}$
$(0 \leftarrow 1[1], 0)\mathcal{S}$
$0 \leftarrow 1[0]$
$(0 \leftarrow 1[0], 2)\mathcal{R}$
$(0 \leftarrow 1[0], 1)\mathcal{M}$
$(0 \leftarrow 1[0], 0)\mathcal{S}$
$0 \leftarrow 0[1]$
$(0 \leftarrow 0[1], 2)\mathcal{M}$
$(0 \leftarrow 0[1], 1)\mathcal{M}$
$(0 \leftarrow 0[1], 0)\mathcal{S}$
$0 \leftarrow 0[0]$
$(0 \leftarrow 0[0], 2)\mathcal{M}$
$(0 \leftarrow 0[0], 1)\mathcal{M}$
$(0 \leftarrow 0[0], 0)\mathcal{S}$
2
1
0
$2 \leftarrow 2[1]$
$(2 \leftarrow 2[1], 2)\mathcal{S}$
$(2 \leftarrow 2[1], 1)\mathcal{M}$
$(2 \leftarrow 2[1], 0)\mathcal{M}$
$2 \leftarrow 2[0]$
$(2 \leftarrow 2[0], 2)\mathcal{S}$
$(2 \leftarrow 2[0], 1)\mathcal{M}$
$(2 \leftarrow 2[0], 0)\mathcal{M}$
$2 \leftarrow 1[1]$
$(2 \leftarrow 1[1], 2)\mathcal{S}$
$(2 \leftarrow 1[1], 1)\mathcal{M}$
$(2 \leftarrow 1[1], 0)\mathcal{R}$
$2 \leftarrow 1[0]$
$(2 \leftarrow 1[0], 2)\mathcal{S}$
$(2 \leftarrow 1[0], 1)\mathcal{M}$
$(2 \leftarrow 1[0], 0)\mathcal{R}$
$2 \leftarrow 0[1]$
$(2 \leftarrow 0[1], 2)\mathcal{S}$
$(2 \leftarrow 0[1], 1)\mathcal{R}$
$(2 \leftarrow 0[1], 0)\mathcal{M}$
$2 \leftarrow 0[0]$
$(2 \leftarrow 0[0], 2)\mathcal{S}$
$(2 \leftarrow 0[0], 1)\mathcal{R}$
$(2 \leftarrow 0[0], 0)\mathcal{M}$
$1 \leftarrow 2[1]$
$(1 \leftarrow 2[1], 2)\mathcal{M}$
$(1 \leftarrow 2[1], 1)\mathcal{S}$
$(1 \leftarrow 2[1], 0)\mathcal{R}$
$1 \leftarrow 2[0]$
$(1 \leftarrow 2[0], 2)\mathcal{M}$
$(1 \leftarrow 2[0], 1)\mathcal{S}$
$(1 \leftarrow 2[0], 0)\mathcal{R}$
$1 \leftarrow 1[1]$
$(1 \leftarrow 1[1], 2)\mathcal{M}$
$(1 \leftarrow 1[1], 1)\mathcal{S}$
$(1 \leftarrow 1[1], 0)\mathcal{M}$
$1 \leftarrow 1[0]$
$(1 \leftarrow 1[0], 2)\mathcal{M}$
$(1 \leftarrow 1[0], 1)\mathcal{S}$
$(1 \leftarrow 1[0], 0)\mathcal{M}$
$1 \leftarrow 0[1]$
$(1 \leftarrow 0[1], 2)\mathcal{R}$
$(1 \leftarrow 0[1], 1)\mathcal{S}$
$(1 \leftarrow 0[1], 0)\mathcal{M}$
$1 \leftarrow 0[0]$
$(1 \leftarrow 0[0], 2)\mathcal{R}$
$(1 \leftarrow 0[0], 1)\mathcal{S}$
$(1 \leftarrow 0[0], 0)\mathcal{M}$
$0 \leftarrow 2[1]$
$(0 \leftarrow 2[1], 2)\mathcal{M}$
$(0 \leftarrow 2[1], 1)\mathcal{R}$
$(0 \leftarrow 2[1], 0)\mathcal{S}$
$0 \leftarrow 2[0]$
$(0 \leftarrow 2[0], 2)\mathcal{M}$
$(0 \leftarrow 2[0], 1)\mathcal{R}$
$(0 \leftarrow 2[0], 0)\mathcal{S}$
$0 \leftarrow 1[1]$
$(0 \leftarrow 1[1], 2)\mathcal{R}$
$(0 \leftarrow 1[1], 1)\mathcal{M}$
$(0 \leftarrow 1[1], 0)\mathcal{S}$
$0 \leftarrow 1[0]$
$(0 \leftarrow 1[0], 2)\mathcal{R}$
$(0 \leftarrow 1[0], 1)\mathcal{M}$
$(0 \leftarrow 1[0], 0)\mathcal{S}$
$0 \leftarrow 0[1]$
$(0 \leftarrow 0[1], 2)\mathcal{M}$
$(0 \leftarrow 0[1], 1)\mathcal{M}$
$(0 \leftarrow 0[1], 0)\mathcal{S}$
$0 \leftarrow 0[0]$
$(0 \leftarrow 0[0], 2)\mathcal{M}$
$(0 \leftarrow 0[0], 1)\mathcal{M}$
$(0 \leftarrow 0[0], 0)\mathcal{S}$
2
1
0

Path:

Virus 1

$1 \leftarrow 1[0]$ (stasis)

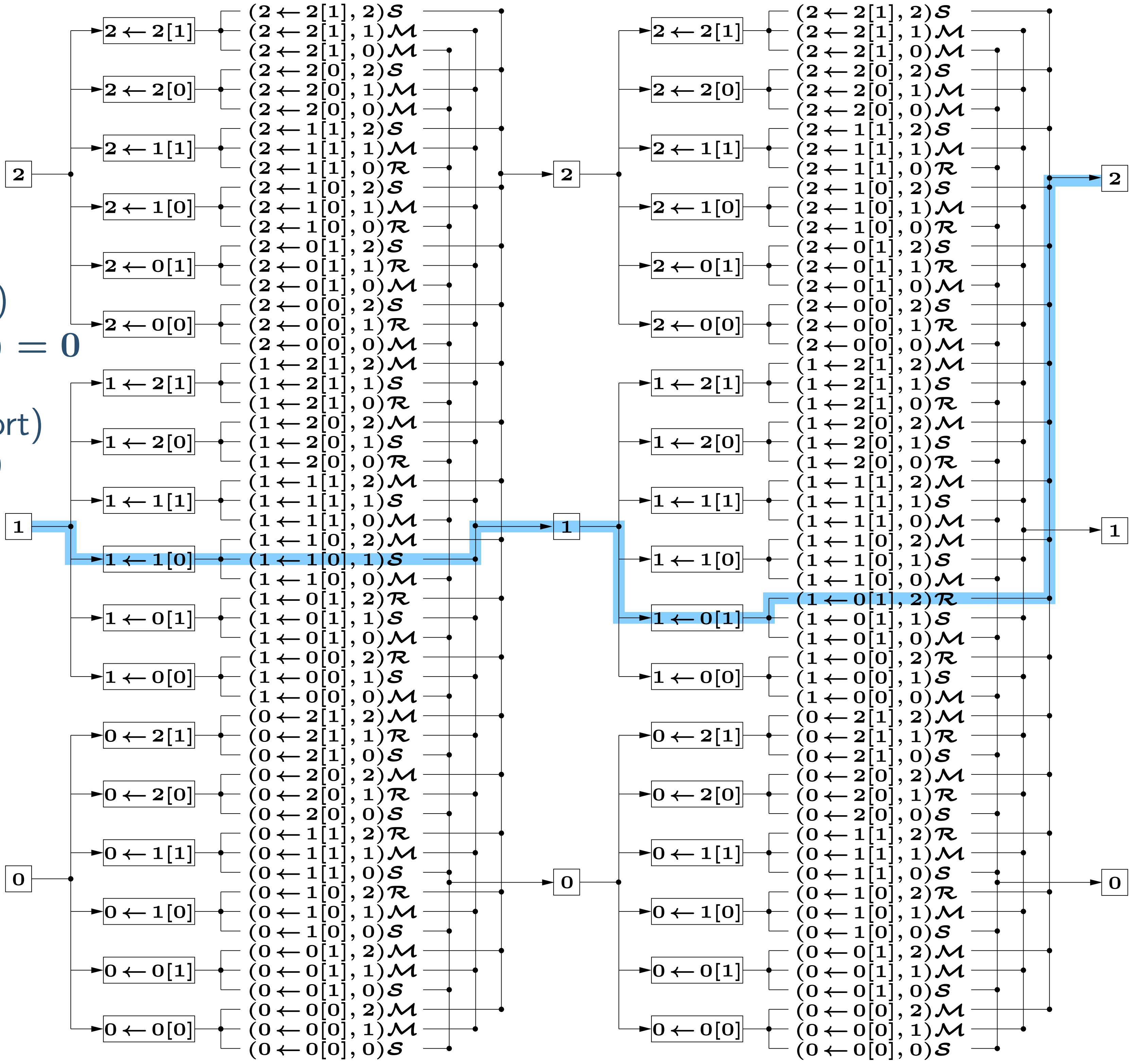
$\mathcal{W}(1 \leftarrow 1[0], 1) = 0$

Virus 1

$1 \leftarrow 0[1]$ (reassort)

$\mathcal{W}(1 \leftarrow 0[1], 2)$

Virus 2



2
1
0
$2 \leftarrow 2[1]$
$2 \leftarrow 2[0]$
$2 \leftarrow 1[1]$
$2 \leftarrow 1[0]$
$2 \leftarrow 0[1]$
$2 \leftarrow 0[0]$
$1 \leftarrow 2[1]$
$1 \leftarrow 2[0]$
$1 \leftarrow 1[1]$
$1 \leftarrow 1[0]$
$1 \leftarrow 0[1]$
$1 \leftarrow 0[0]$
$0 \leftarrow 2[1]$
$0 \leftarrow 2[0]$
$0 \leftarrow 1[1]$
$0 \leftarrow 1[0]$
$0 \leftarrow 0[1]$
$0 \leftarrow 0[0]$
$(2 \leftarrow 2[1], 2)\mathcal{S}$
$(2 \leftarrow 2[1], 1)\mathcal{M}$
$(2 \leftarrow 2[1], 0)\mathcal{M}$
$(2 \leftarrow 2[0], 2)\mathcal{S}$
$(2 \leftarrow 2[0], 1)\mathcal{M}$
$(2 \leftarrow 2[0], 0)\mathcal{M}$
$(2 \leftarrow 1[1], 2)\mathcal{S}$
$(2 \leftarrow 1[1], 1)\mathcal{M}$
$(2 \leftarrow 1[1], 0)\mathcal{R}$
$(2 \leftarrow 1[0], 2)\mathcal{S}$
$(2 \leftarrow 1[0], 1)\mathcal{M}$
$(2 \leftarrow 1[0], 0)\mathcal{R}$
$(2 \leftarrow 0[1], 2)\mathcal{S}$
$(2 \leftarrow 0[1], 1)\mathcal{R}$
$(2 \leftarrow 0[1], 0)\mathcal{M}$
$(2 \leftarrow 0[0], 2)\mathcal{S}$
$(2 \leftarrow 0[0], 1)\mathcal{R}$
$(2 \leftarrow 0[0], 0)\mathcal{M}$
$(1 \leftarrow 2[1], 2)\mathcal{M}$
$(1 \leftarrow 2[1], 1)\mathcal{S}$
$(1 \leftarrow 2[1], 0)\mathcal{R}$
$(1 \leftarrow 2[0], 2)\mathcal{M}$
$(1 \leftarrow 2[0], 1)\mathcal{S}$
$(1 \leftarrow 2[0], 0)\mathcal{R}$
$(1 \leftarrow 1[1], 2)\mathcal{M}$
$(1 \leftarrow 1[1], 1)\mathcal{S}$
$(1 \leftarrow 1[1], 0)\mathcal{M}$
$(1 \leftarrow 1[0], 2)\mathcal{M}$
$(1 \leftarrow 1[0], 1)\mathcal{S}$
$(1 \leftarrow 1[0], 0)\mathcal{M}$
$(1 \leftarrow 0[1], 2)\mathcal{R}$
$(1 \leftarrow 0[1], 1)\mathcal{S}$
$(1 \leftarrow 0[1], 0)\mathcal{M}$
$(1 \leftarrow 0[0], 2)\mathcal{R}$
$(1 \leftarrow 0[0], 1)\mathcal{S}$
$(1 \leftarrow 0[0], 0)\mathcal{M}$
$(0 \leftarrow 2[1], 2)\mathcal{M}$
$(0 \leftarrow 2[1], 1)\mathcal{R}$
$(0 \leftarrow 2[1], 0)\mathcal{S}$
$(0 \leftarrow 2[0], 2)\mathcal{M}$
$(0 \leftarrow 2[0], 1)\mathcal{R}$
$(0 \leftarrow 2[0], 0)\mathcal{S}$
$(0 \leftarrow 1[1], 2)\mathcal{R}$
$(0 \leftarrow 1[1], 1)\mathcal{M}$
$(0 \leftarrow 1[1], 0)\mathcal{S}$
$(0 \leftarrow 1[0], 2)\mathcal{R}$
$(0 \leftarrow 1[0], 1)\mathcal{M}$
$(0 \leftarrow 1[0], 0)\mathcal{S}$
$(0 \leftarrow 0[1], 2)\mathcal{M}$
$(0 \leftarrow 0[1], 1)\mathcal{M}$
$(0 \leftarrow 0[1], 0)\mathcal{S}$
$(0 \leftarrow 0[0], 2)\mathcal{M}$
$(0 \leftarrow 0[0], 1)\mathcal{M}$
$(0 \leftarrow 0[0], 0)\mathcal{S}$
2
1
0
$2 \leftarrow 2[1]$
$2 \leftarrow 2[0]$
$2 \leftarrow 1[1]$
$2 \leftarrow 1[0]$
$2 \leftarrow 0[1]$
$2 \leftarrow 0[0]$
$1 \leftarrow 2[1]$
$1 \leftarrow 2[0]$
$1 \leftarrow 1[1]$
$1 \leftarrow 1[0]$
$1 \leftarrow 0[1]$
$1 \leftarrow 0[0]$
$0 \leftarrow 2[1]$
$0 \leftarrow 2[0]$
$0 \leftarrow 1[1]$
$0 \leftarrow 1[0]$
$0 \leftarrow 0[1]$
$0 \leftarrow 0[0]$
$(2 \leftarrow 2[1], 2)\mathcal{S}$
$(2 \leftarrow 2[1], 1)\mathcal{M}$
$(2 \leftarrow 2[1], 0)\mathcal{M}$
$(2 \leftarrow 2[0], 2)\mathcal{S}$
$(2 \leftarrow 2[0], 1)\mathcal{M}$
$(2 \leftarrow 2[0], 0)\mathcal{M}$
$(2 \leftarrow 1[1], 2)\mathcal{S}$
$(2 \leftarrow 1[1], 1)\mathcal{M}$
$(2 \leftarrow 1[1], 0)\mathcal{R}$
$(2 \leftarrow 1[0], 2)\mathcal{S}$
$(2 \leftarrow 1[0], 1)\mathcal{M}$
$(2 \leftarrow 1[0], 0)\mathcal{R}$
$(2 \leftarrow 0[1], 2)\mathcal{S}$
$(2 \leftarrow 0[1], 1)\mathcal{R}$
$(2 \leftarrow 0[1], 0)\mathcal{M}$
$(2 \leftarrow 0[0], 2)\mathcal{S}$
$(2 \leftarrow 0[0], 1)\mathcal{R}$
$(2 \leftarrow 0[0], 0)\mathcal{M}$
$(1 \leftarrow 2[1], 2)\mathcal{M}$
$(1 \leftarrow 2[1], 1)\mathcal{S}$
$(1 \leftarrow 2[1], 0)\mathcal{R}$
$(1 \leftarrow 2[0], 2)\mathcal{M}$
$(1 \leftarrow 2[0], 1)\mathcal{S}$
$(1 \leftarrow 2[0], 0)\mathcal{R}$
$(1 \leftarrow 1[1], 2)\mathcal{M}$
$(1 \leftarrow 1[1], 1)\mathcal{S}$
$(1 \leftarrow 1[1], 0)\mathcal{M}$
$(1 \leftarrow 1[0], 2)\mathcal{M}$
$(1 \leftarrow 1[0], 1)\mathcal{S}$
$(1 \leftarrow 1[0], 0)\mathcal{M}$
$(1 \leftarrow 0[1], 2)\mathcal{R}$
$(1 \leftarrow 0[1], 1)\mathcal{S}$
$(1 \leftarrow 0[1], 0)\mathcal{M}$
$(1 \leftarrow 0[0], 2)\mathcal{R}$
$(1 \leftarrow 0[0], 1)\mathcal{S}$
$(1 \leftarrow 0[0], 0)\mathcal{M}$
$(0 \leftarrow 2[1], 2)\mathcal{M}$
$(0 \leftarrow 2[1], 1)\mathcal{R}$
$(0 \leftarrow 2[1], 0)\mathcal{S}$
$(0 \leftarrow 2[0], 2)\mathcal{M}$
$(0 \leftarrow 2[0], 1)\mathcal{R}$
$(0 \leftarrow 2[0], 0)\mathcal{S}$
$(0 \leftarrow 1[1], 2)\mathcal{R}$
$(0 \leftarrow 1[1], 1)\mathcal{M}$
$(0 \leftarrow 1[1], 0)\mathcal{S}$
$(0 \leftarrow 1[0], 2)\mathcal{R}$
$(0 \leftarrow 1[0], 1)\mathcal{M}$
$(0 \leftarrow 1[0], 0)\mathcal{S}$
$(0 \leftarrow 0[1], 2)\mathcal{M}$
$(0 \leftarrow 0[1], 1)\mathcal{M}$
$(0 \leftarrow 0[1], 0)\mathcal{S}$
$(0 \leftarrow 0[0], 2)\mathcal{M}$
$(0 \leftarrow 0[0], 1)\mathcal{M}$
$(0 \leftarrow 0[0], 0)\mathcal{S}$
2
1
0

Key features of the XMT

- Hardware support for 128 threads per processor

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime
- Powerful interconnect⇒Flat shared memory

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime
- Powerful interconnect⇒Flat shared memory
- Full/Empty bits ⇒Word-level synchronization

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime
- Powerful interconnect⇒Flat shared memory
- Full/Empty bits ⇒Word-level synchronization
  - Existing serial C or C++ code can be parallelized with minor effort

Key features of the XMT

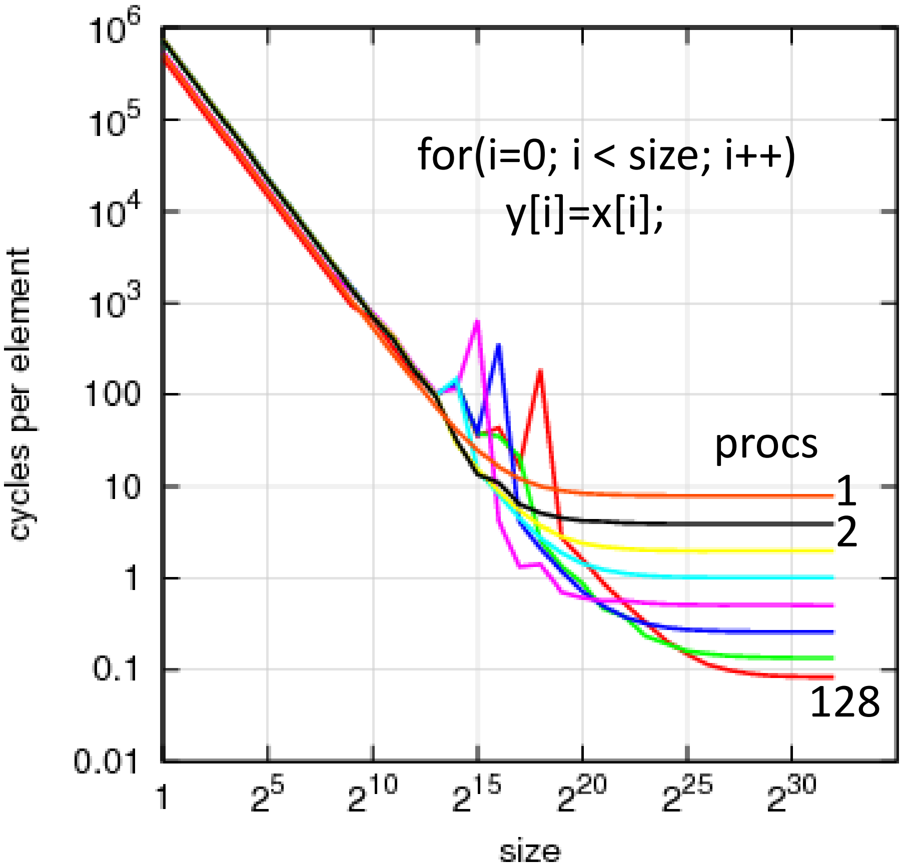
- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime
- Powerful interconnect⇒Flat shared memory
- Full/Empty bits ⇒Word-level synchronization
    - Existing serial C or C++ code can be parallelized with minor effort
    - No need for explicit parallel programming constructs, load balancing, partitioning, mapping, mesg. passing

Key features of the XMT

- Hardware support for 128 threads per processor
- 128 proc. machine has hardware for 128^2=16384 thr.
- Zero overhead switching between threads
- Total threads essentially unlimited–managed by runtime
- Powerful interconnect⇒Flat shared memory
- Full/Empty bits ⇒Word-level synchronization
    - Existing serial C or C++ code can be parallelized with minor effort
    - No need for explicit parallel programming constructs, load balancing, partitioning, mapping, mesg. passing
    - Instead of dividing up the *problem*, the *machine* is divided into small agile units that can self-schedule with little overhead

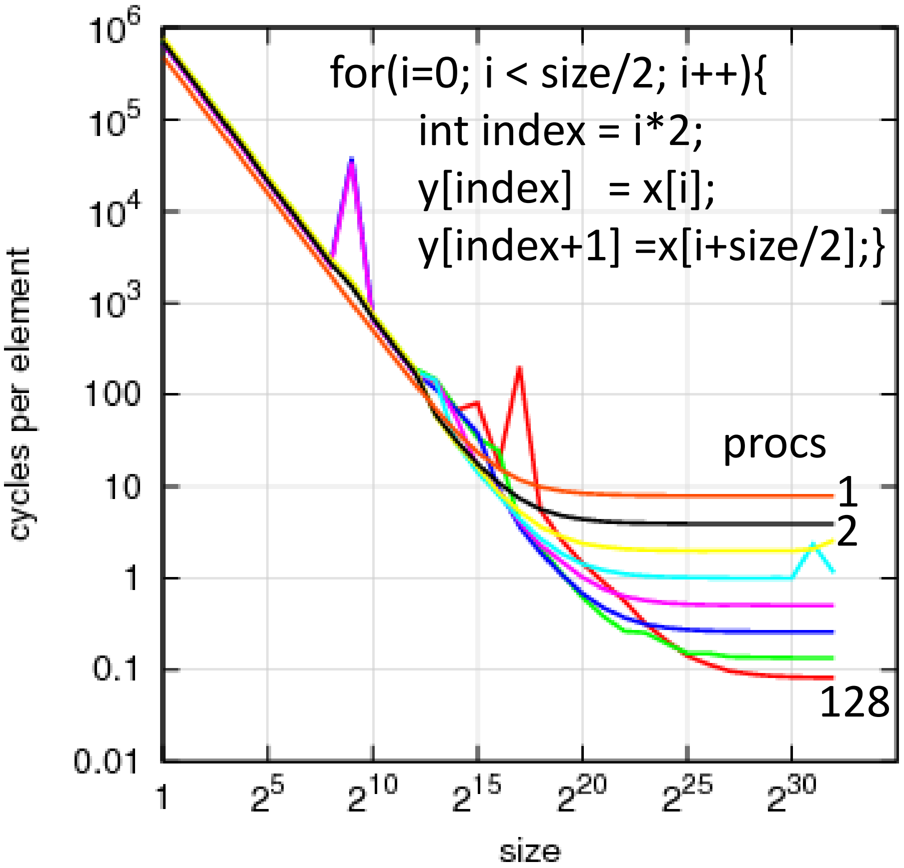
Flat Shared memory: insensitive to access patterns

Linear Access



```
for(i=0; i < size; i++)
     y[i]=x[i];
```

Perfect Shuffle



```
for(i=0; i < size/2; i++){
     int index = i*2;
     y[index]   = x[i];
     y[index+1] =x[i+size/2];}
```

➔ **NOTE: log-log scales (here and in later slides)**

Example Code

```
void labelVirusesToEvents(stage t){
   int i;
   #pragma mta assert parallel
   for(i=0; i<numViruses; i++){
      int j=i, k;
      #pragma mta assert parallel
      for(k=0; k<numViruses; k++){
         int s;
         for(s=0; s<numSegments; s++){
            int temp = V[t][i] + costVR(i,j,k,s);
            int myR = readfe(&R[t][j][k][s]);
            if(temp<myR){
               myR = temp;
               whichVR[t][j][k][s] = i;
            }
            writeef(&R[t][j][k][s],myR);
         }
      }
   }
}
```

Performance

- 32 stages on 128 proc. Cray XMT at Pacific Northwest National Laboratory (35 hrs)

Performance

- 32 stages on 128 proc. Cray XMT at Pacific Northwest National Laboratory (35 hrs)
- Loop optimizations and upgrade to XMT-2 would have reduced time to 17 hrs.

Performance

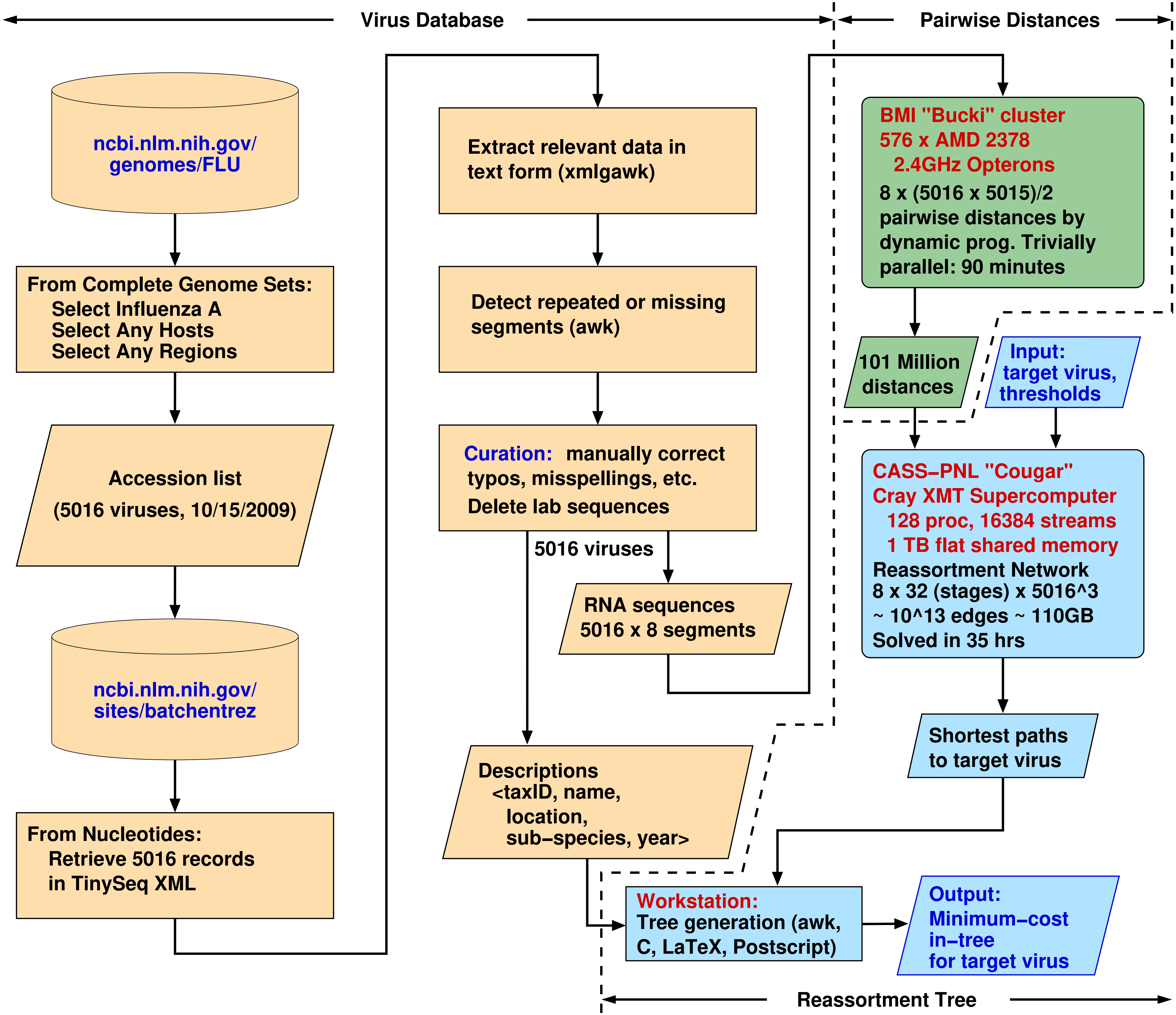
- 32 stages on 128 proc. Cray XMT at Pacific Northwest National Laboratory (35 hrs)
- Loop optimizations and upgrade to XMT-2 would have reduced time to 17 hrs.
- Edit costs (~~embarassingly~~ trivially parallel): 90 min on conventional cluster (576 node AMD 2378 2.4GHz Opterons.)

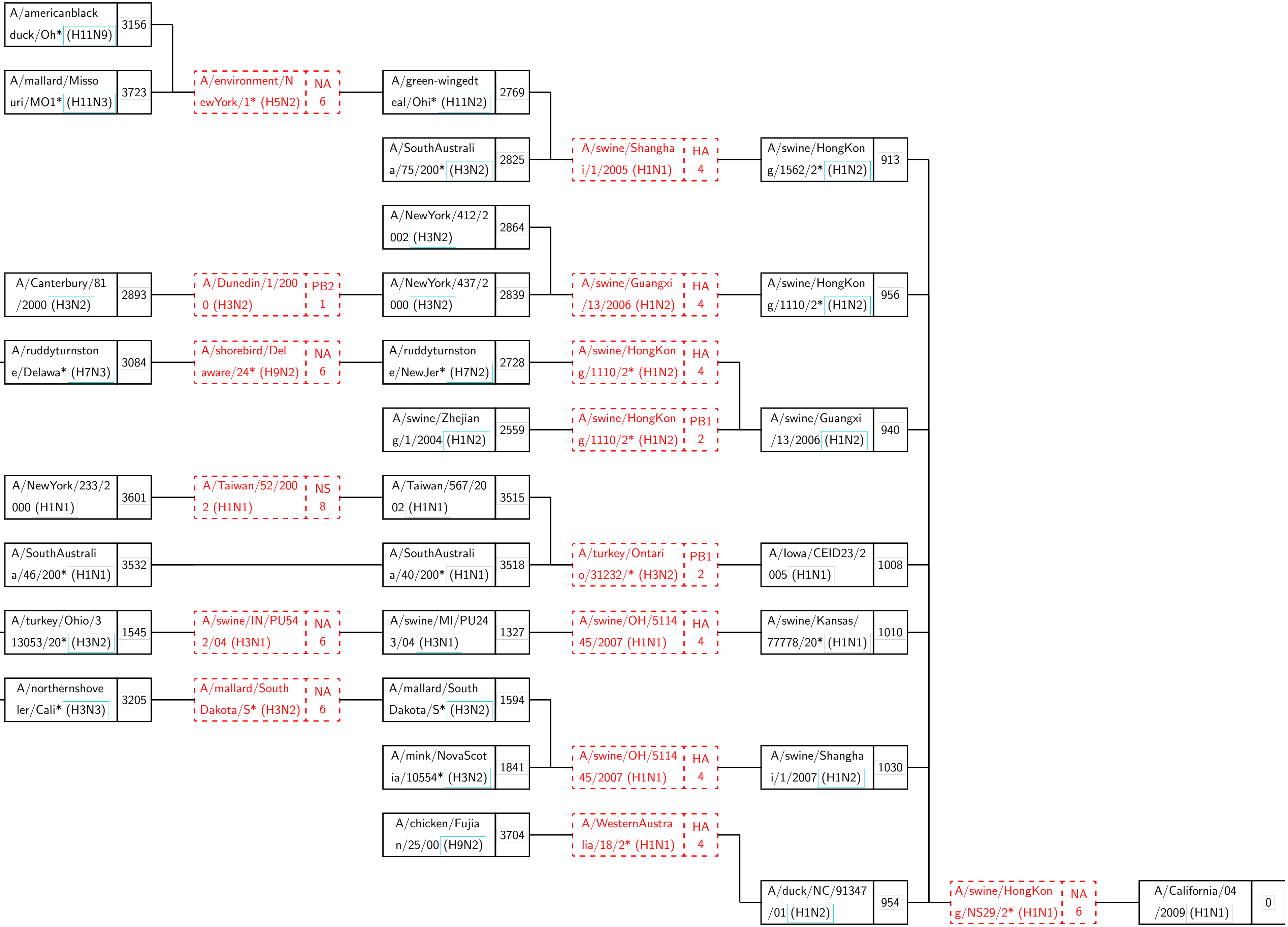
Performance

- 32 stages on 128 proc. Cray XMT at Pacific Northwest National Laboratory (35 hrs)
- Loop optimizations and upgrade to XMT-2 would have reduced time to 17 hrs.
- Edit costs (~~embarassingly~~ trivially parallel): 90 min on conventional cluster (576 node AMD 2378 2.4GHz Opterons.)
- Paths can be combined to form in- or out-trees
  In-trees give shortest paths from all ancestors to a specific virus.
  Out-trees give shortest paths from a specific virus to all descendents.

Performance

- 32 stages on 128 proc. Cray XMT at Pacific Northwest National Laboratory (35 hrs)
- Loop optimizations and upgrade to XMT-2 would have reduced time to 17 hrs.
- Edit costs (~~embarassingly~~ trivially parallel): 90 min on conventional cluster (576 node AMD 2378 2.4GHz Opterons.)
- Paths can be combined to form in- or out-trees In-trees give shortest paths from all ancestors to a specific virus.
  Out-trees give shortest paths from a specific virus to all descendents.
- Computation of each out-tree: 10 min


Virus Database
Pairwise Distances
ncbi.nlm.nih.gov/ genomes/FLU
From Complete Genome Sets:
Select Influenza A
Select Any Hosts
Select Any Regions
Accession list
(5016 viruses, 10/15/2009)
ncbi.nlm.nih.gov/ sites/batchentrez
From Nucleotides:
Retrieve 5016 records
in TinySeq XML
Extract relevant data in text form (xmlgawk)
Detect repeated or missing segments (awk)
Curation: manually correct typos, misspellings, etc.
Delete lab sequences
5016 viruses
RNA sequences
5016 x 8 segments
Descriptions
<taxID, name,
location,
sub−species, year>
BMI "Bucki" cluster
576 x AMD 2378
2.4GHz Opterons
8 x (5016 x 5015)/2
pairwise distances by
dynamic prog. Trivially
parallel: 90 minutes
101 Million distances
Input:
target virus,
thresholds
CASS−PNL "Cougar"
Cray XMT Supercomputer
128 proc, 16384 streams
1 TB flat shared memory
Reassortment Network
8 x 32 (stages) x 5016^3
~ 10^13 edges ~ 110GB
Solved in 35 hrs
Shortest paths
to target virus
Workstation:
Tree generation (awk,
C, LaTeX, Postscript)
Output:
Minimum−cost
in−tree
for target virus
Reassortment Tree


A/americanblack duck/Oh* (H11N9) 3156
A/mallard/Missouri/MO1* (H11N3) 3723
A/environment/NewYork/1* (H5N2) NA 6
A/green-wingedteal/Ohi* (H11N2) 2769
A/SouthAustralia/75/200* (H3N2) 2825
A/swine/Shanghai/1/2005 (H1N1) HA 4
A/swine/HongKong/1562/2* (H1N2) 913
A/NewYork/412/2002 (H3N2) 2864
A/Canterbury/81/2000 (H3N2) 2893
A/Dunedin/1/2000 (H3N2) PB2 1
A/NewYork/437/2000 (H3N2) 2839
A/swine/Guangxi/13/2006 (H1N2) HA 4
A/swine/HongKong/1110/2* (H1N2) 956
A/ruddyturnstone/Delawa* (H7N3) 3084
A/shorebird/Delaware/24* (H9N2) NA 6
A/ruddyturnstone/NewJer* (H7N2) 2728
A/swine/HongKong/1110/2* (H1N2) HA 4
A/swine/Zhejiang/1/2004 (H1N2) 2559
A/swine/HongKong/1110/2* (H1N2) PB1 2
A/swine/Guangxi/13/2006 (H1N2) 940
A/NewYork/233/2000 (H1N1) 3601
A/Taiwan/52/2002 (H1N1) NS 8
A/Taiwan/567/2002 (H1N1) 3515
A/SouthAustralia/46/200* (H1N1) 3532
A/SouthAustralia/40/200* (H1N1) 3518
A/turkey/Ontario/31232/* (H3N2) PB1 2
A/Iowa/CEID23/2005 (H1N1) 1008
A/turkey/Ohio/313053/20* (H3N2) 1545
A/swine/IN/PU542/04 (H3N1) NA 6
A/swine/MI/PU243/04 (H3N1) 1327
A/swine/OH/511445/2007 (H1N1) HA 4
A/swine/Kansas/77778/20* (H1N1) 1010
A/northernshoveler/Cali* (H3N3) 3205
A/mallard/SouthDakota/S* (H3N2) NA 6
A/mallard/SouthDakota/S* (H3N2) 1594
A/mink/NovaScotia/10554* (H3N2) 1841
A/swine/OH/511445/2007 (H1N1) HA 4
A/swine/Shanghai/1/2007 (H1N2) 1030
A/chicken/Fujian/25/00 (H9N2) 3704
A/WesternAustralia/18/2* (H1N1) HA 4
A/duck/NC/91347/01 (H1N2) 954
A/swine/HongKong/NS29/2* (H1N1) NA 6
A/California/04/2009 (H1N1) 0

Results: 6 Bottleneck Viruses

A set through which most paths pass before reaching S-OIV **A/California/04/2009**

Numbers of paths through bottleneck viruses

1	A/swine/Shanghai/1/2007	292
2	A/swine/Guangxi/13/2006	1252
3	A/swine/HongKong/1110/2006	199
4	A/swine/HongKong/1562/2005	919
5	A/swine/Kansas/77778/2007	736
6	A/Iowa/CEID23/2005	202
	Paths through bottleneck viruses	3600
	Total paths in tree	3926

Bottleneck Viruses

Pairwise distances *between* bottleneck viruses.

1	2	3	4	5	6	
0	554	541	487	1706	1662	1
	0	473	426	1735	1753	2
		0	349	1643	1682	3
			0	1625	1666	4
				0	654	5
					0	6

Molecular biology of bottleneck viruses needs more investigation!

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.
- **THE BIG QUESTIONS:**

Overview
Viruses
Evolution
Covid-19
Phylogenetic Trees
Tree/web of Life
Antigenic Shift
Results
Model
Layered Graph
Features
Example Code
Performance
Workflow
Tree
Bottleneck Viruses
Bottleneck Viruses
Extensions
Future

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.
- **THE BIG QUESTIONS:**
  - Where are the recombinations?

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.
- **THE BIG QUESTIONS:**
  - Where are the recombinations?
  - What animals?

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.
- **THE BIG QUESTIONS:**
  - Where are the recombinations?
  - What animals?
  - When?

Extend approach to Covid-19

- Treat each protein as a segment–reduces to our influenza approach, OR
- Use $O(n^2)$ dynamic programming variants to find locally optimal edit distances. Expensive time-wise but trivially parallel.
- Create layered graph.
- Dynamic Programming difficult without the XMT but worthwhile, even if it takes days.
- Find lowest cost paths in layered graph.
- **THE BIG QUESTIONS:**
  - Where are the recombinations?
  - What animals?
  - When?
  - Where?

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses

Future Work

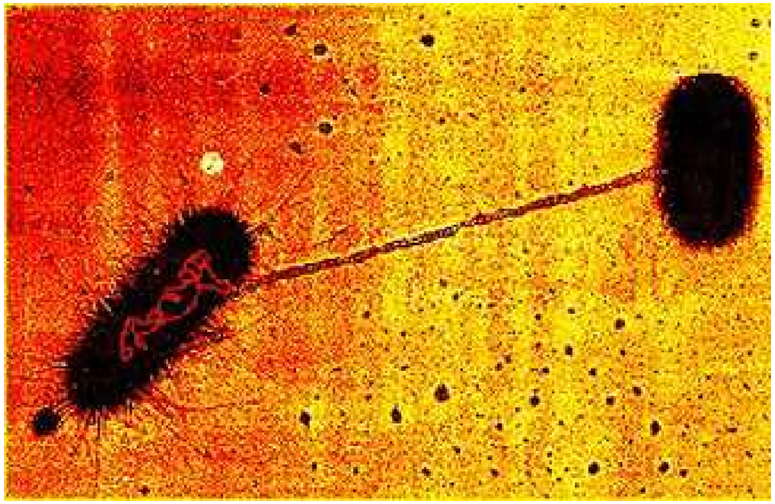
- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)
- Study other Horizontal Gene Transfer (HGT) phenomena.

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)
- Study other Horizontal Gene Transfer (HGT) phenomena.
- In particular, bacterial conjugation (implicated in development of antibiotic resistance)



Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)
- Study other Horizontal Gene Transfer (HGT) phenomena.
- In particular, bacterial conjugation (implicated in development of antibiotic resistance)
- Refine Path model

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)
- Study other Horizontal Gene Transfer (HGT) phenomena.
- In particular, bacterial conjugation (implicated in development of antibiotic resistance)
- Refine Path model
    - Sum $\longrightarrow$ Bottleneck, or Sum-Bottleneck.

Future Work

- Analyze *all known* influenza-A viruses (not just H1N1 SOIV)
- Extend to other segmented viruses
- Analyze *all known* Coronaviruses (not just Covid-19)
- Study other Horizontal Gene Transfer (HGT) phenomena.
- In particular, bacterial conjugation (implicated in development of antibiotic resistance)
- Refine Path model
  - Sum ⟶ Bottleneck, or Sum-Bottleneck.
  - Incremental changes in input data as new sequences are reported.