Sampling- and Estimation-based Strategies for Data Collection in Wafer-Scale Evolution Simulations

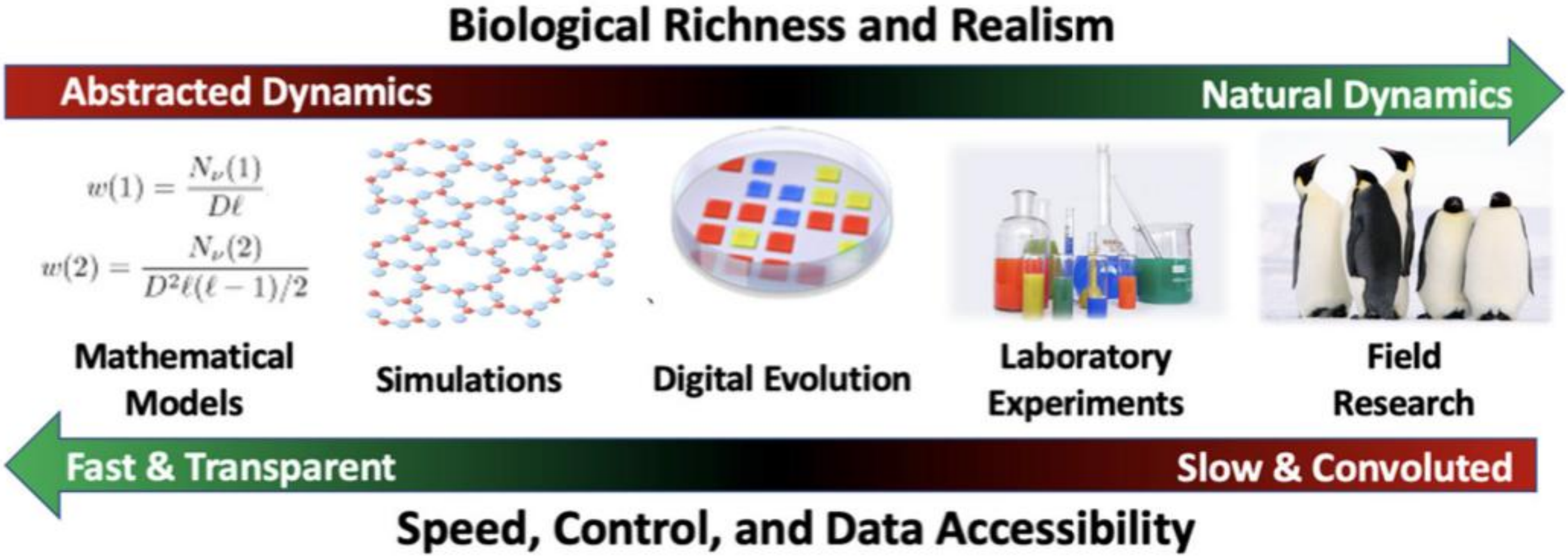
March 5, 2026 @ PSC

Matthew Andres Moreno
Complex Systems, Michigan Institute for Data and AI in Society, Ecology and Evolutionary Biology
University of Michigan

these slides: hopth.ru/gg

Agent-based Evolution & Simulation Scale

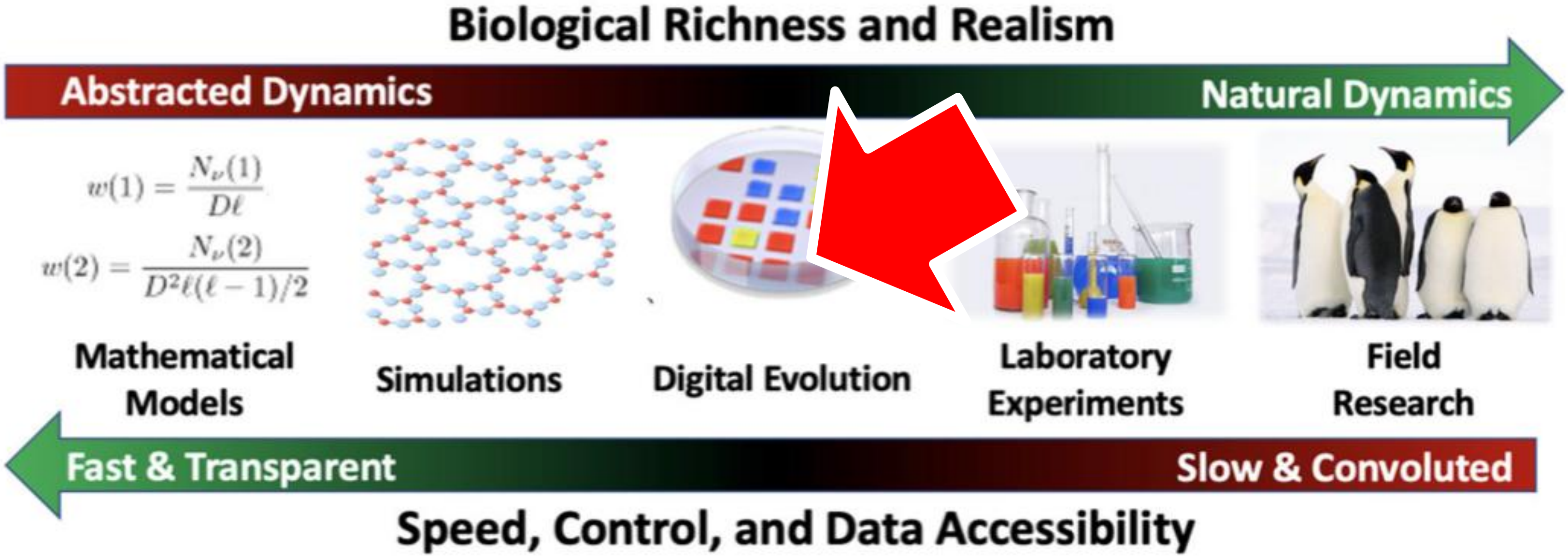
Scale and Agent-based Evolution Simulations



(Dolson and Ofria, 2021)

morenoma@umich.edu

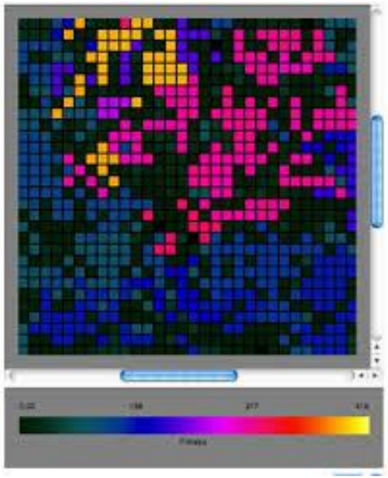
Scale and Agent-based Evolution Simulations



(Dolson and Ofria, 2021)

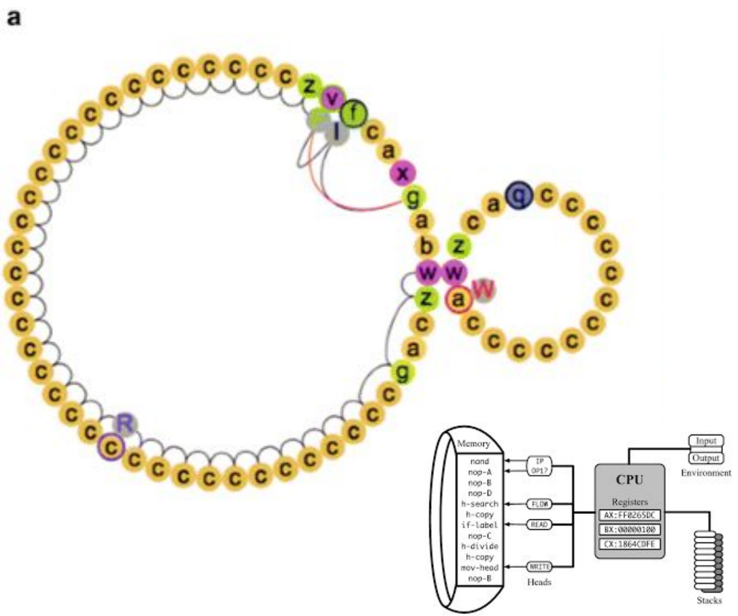
morenoma@umich.edu

Scale and Digital Evolution



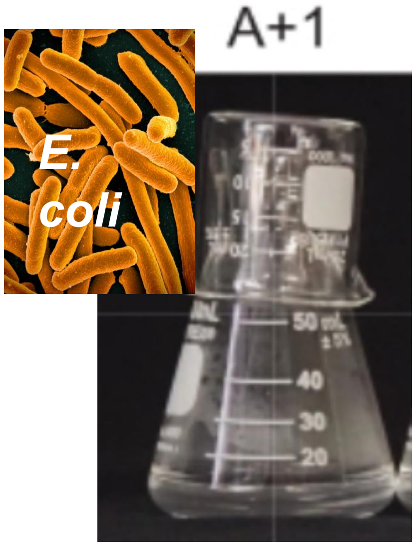
Avida

(Ofria and Wilke, 2009)



morenoma@umich.edu

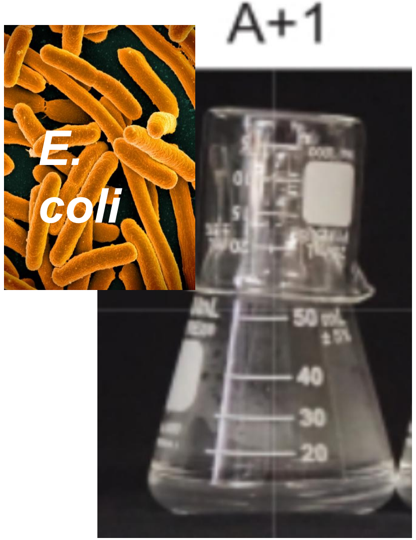
Evolution Models *in vivo* and *in silico*



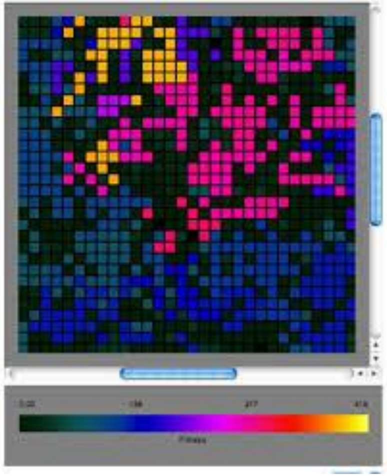
LTEE
(Good et al., 2017)

experimental evolution:
n>1, experimental manipulations

Evolution Models *in vivo* and *in silico*



LTEE
(Good et al., 2017)



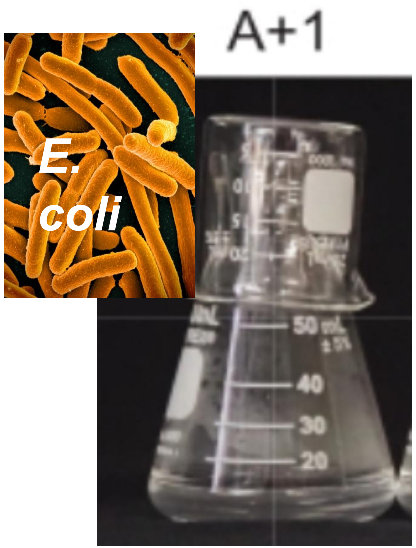
Avida
(Ofria and Wilke, 2009)

experimental evolution:
n>1, experimental manipulations

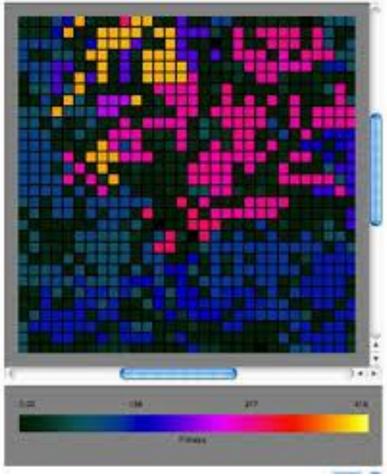
simulation and modeling:
synthesize theory, test sufficiency

morenoma@umich.edu

Evolution Models *in vivo* and *in silico*



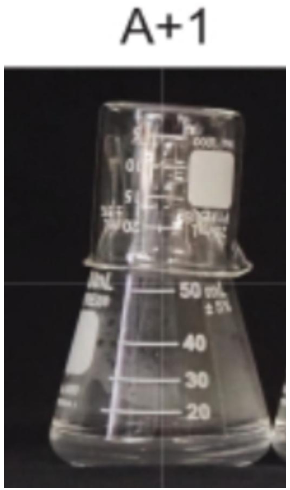
LTEE
(Good et al., 2017)

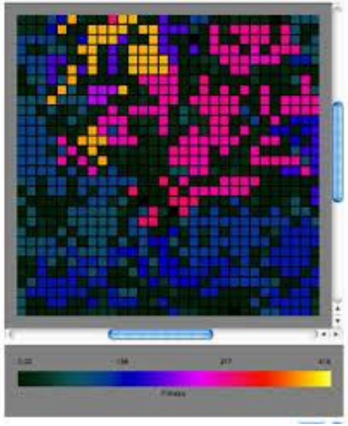


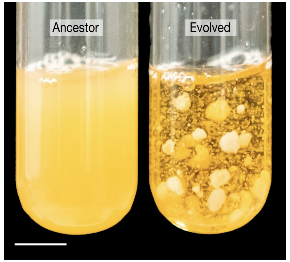
Avida
(Ofria and Wilke, 2009)

morenoma@umich.edu

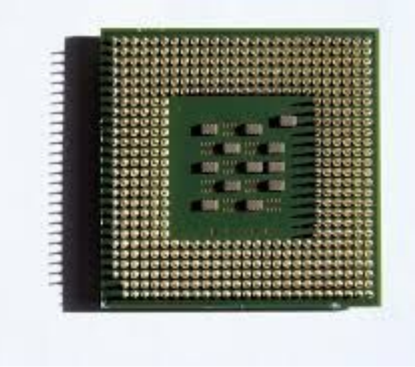
Scale and Digital Evolution





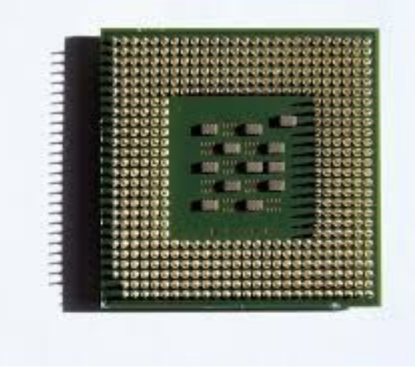


1
processor

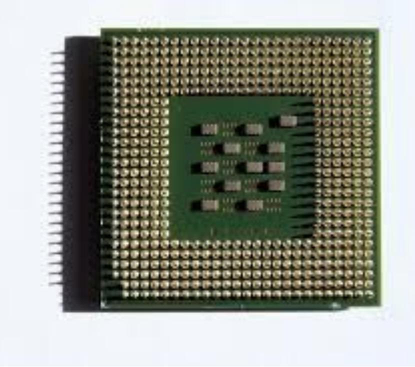


~billion replications/day

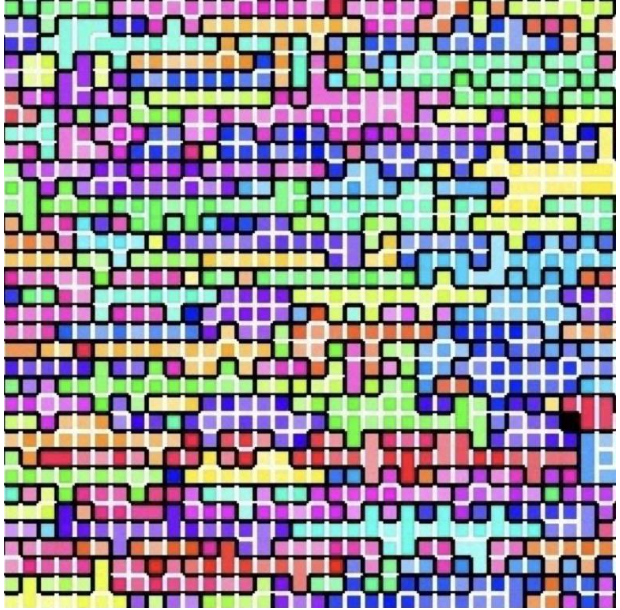
1 processor



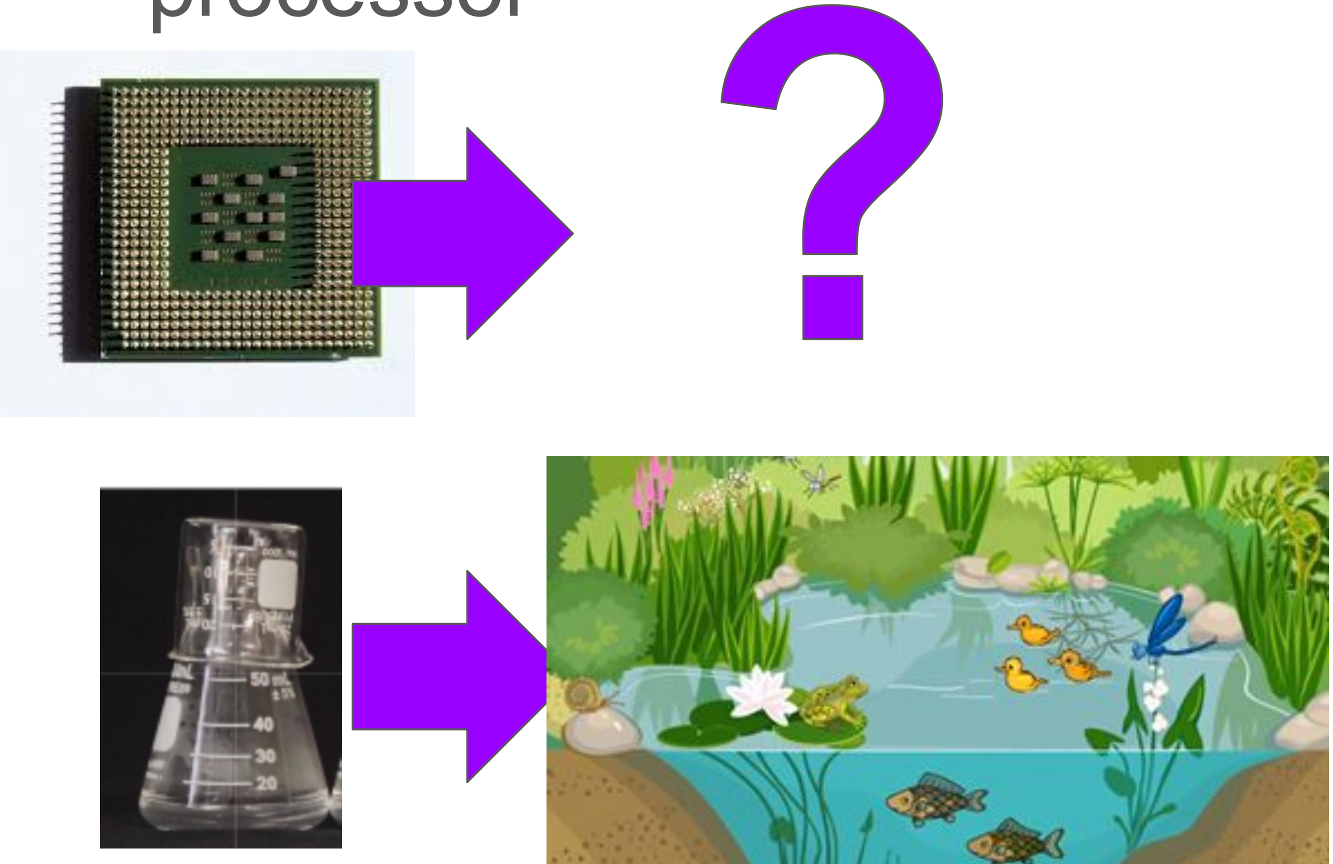
1
processor

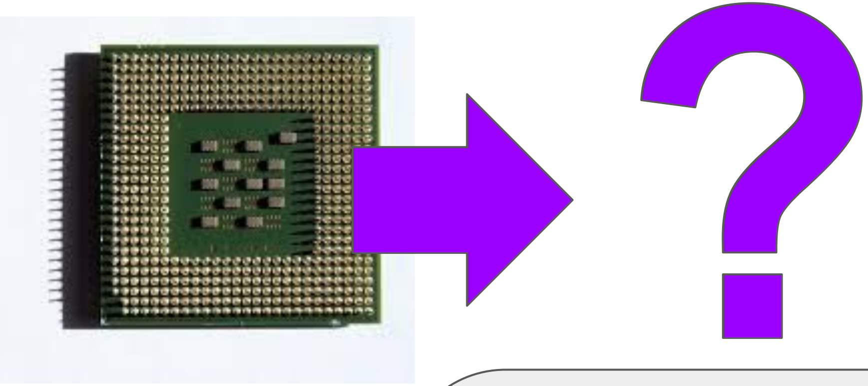


e.g.,
digital
multicell
experiment



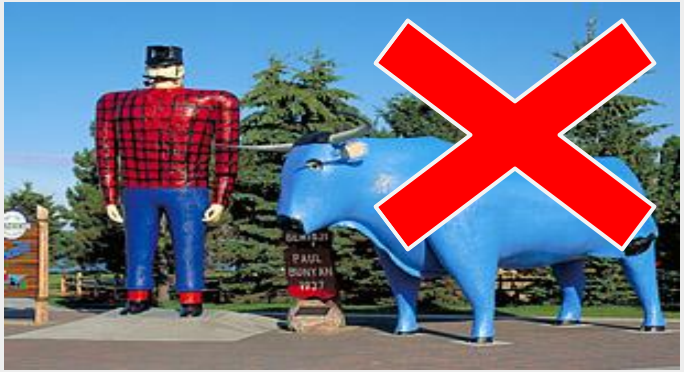
1
processor





Grace Hopper

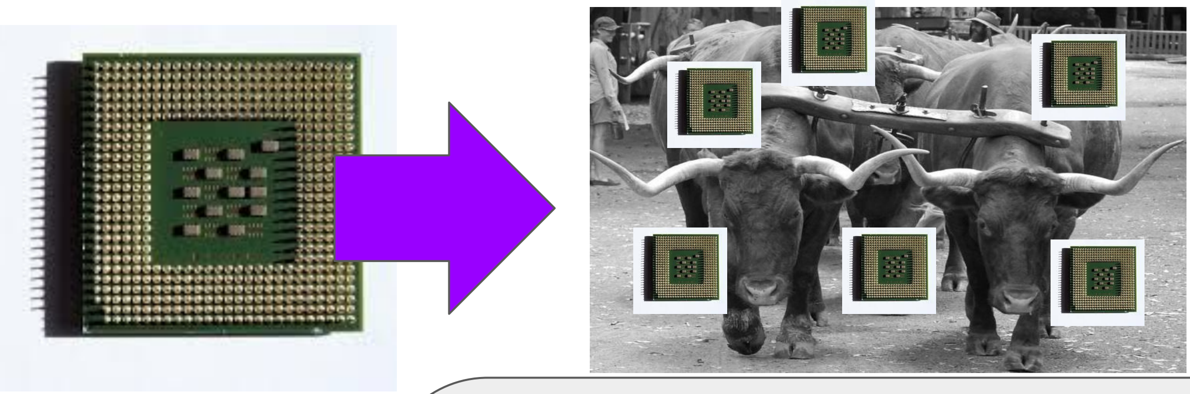
a bigger ox



Nejones1987, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

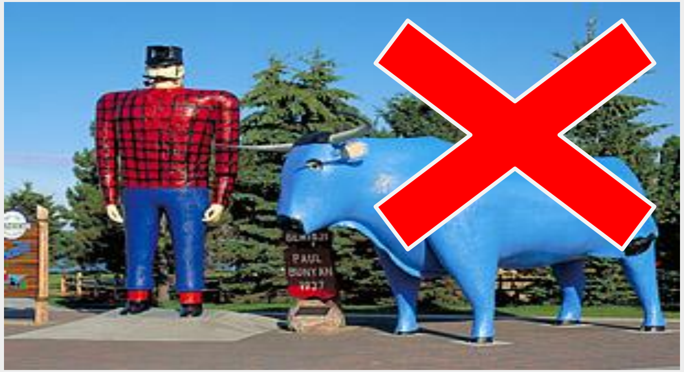
a team of oxen

Cgoodwin, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons



Grace Hopper

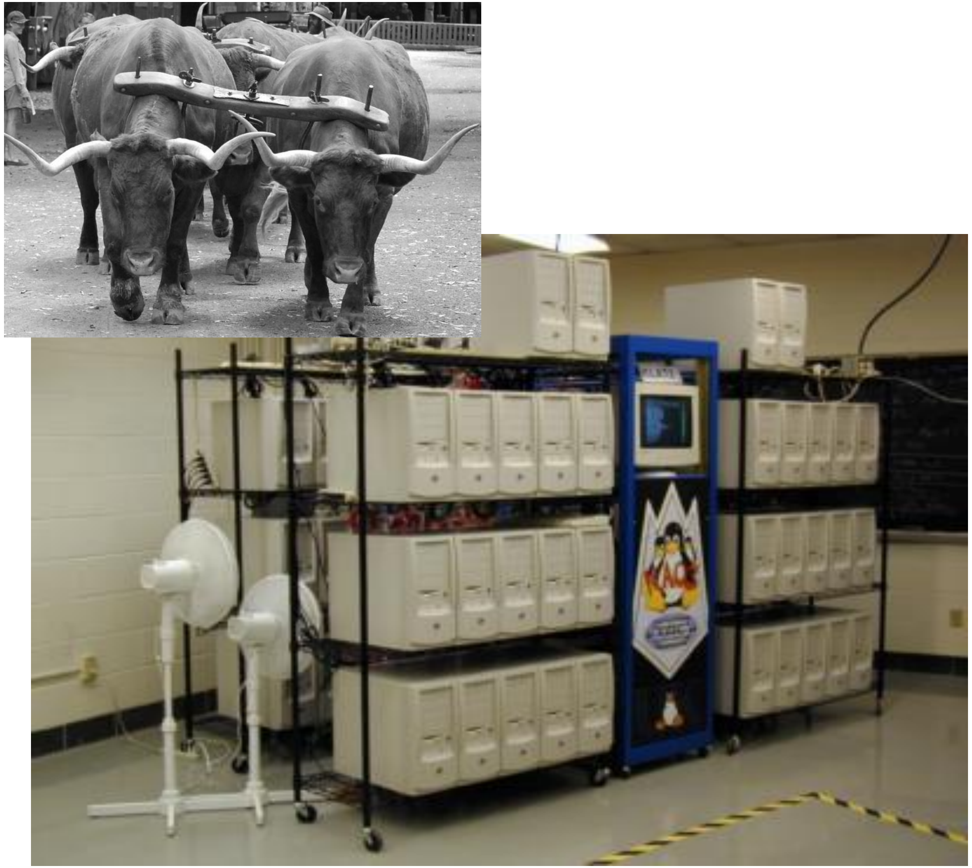
a bigger ox



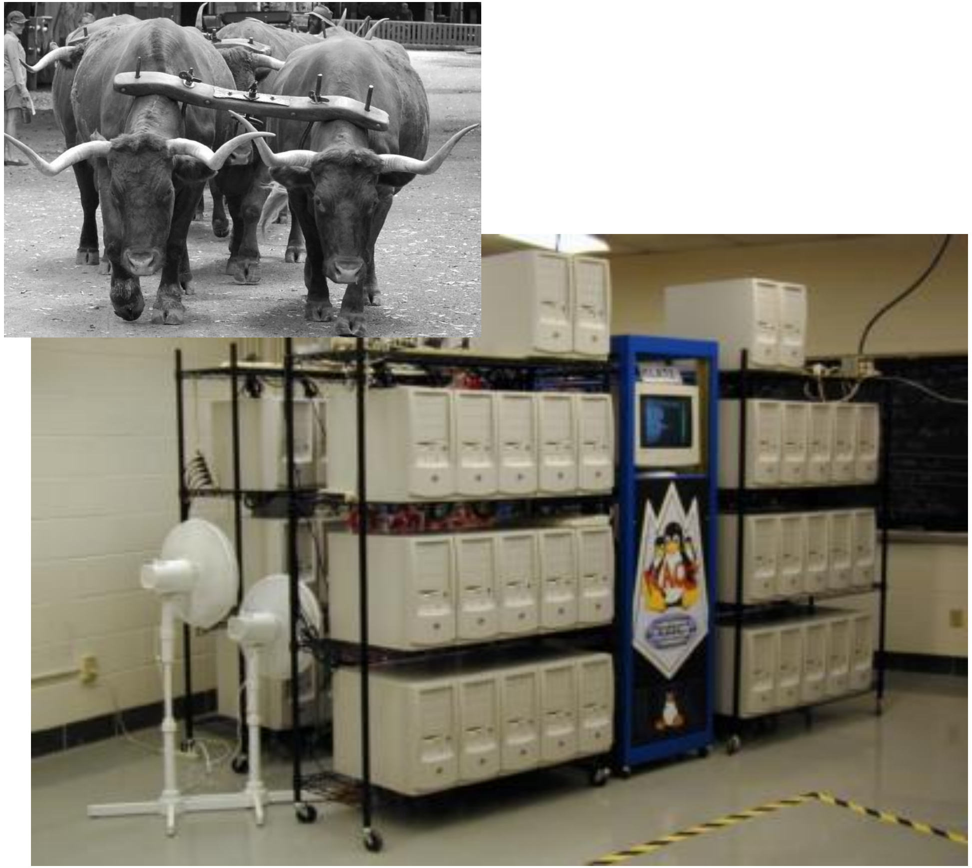
Nejones1987, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

a team of oxen

Cgoodwin, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons



KLAT2 (HANK DIETZ)



KLAT2 (HANK DIETZ)

KLAT2 (HANK DIETZ)

Fugaku

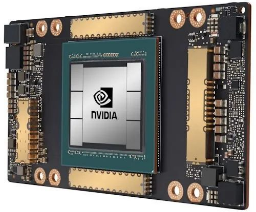
The Library of Congress via Wikimedia Commons

“next-generation”
high-performance computing
(especially AI/ML)

KLAT2 (HANK DIETZ)

The Library of Congress via Wikimedia Commons

KLAT2 (HANK DIETZ)

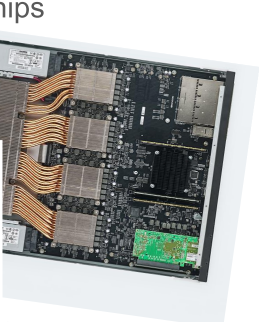


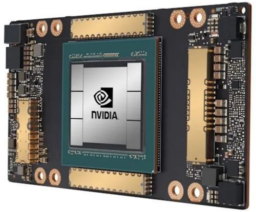
NVIDIA A100 GPU
6,912 CUDA cores

The Library of Congress via Wikimedia Commons

KLAT2 (HANK DIETZ)

Graphcore IPU
1,200 cores per chip
clustered up to 1,024 chips





NVIDIA A100 GPU
6,912 CUDA cores

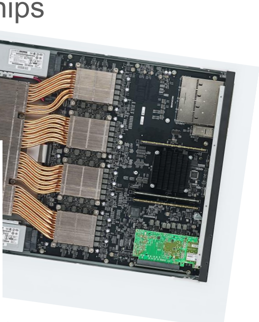
SambaNova RDU

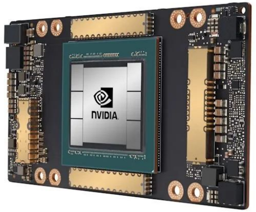
KLAT2 (HANK DIETZ)

The Library of Congress via Wikimedia Commons

Cerebras Wafer-Scale Engine

Graphcore IPU
1,200 cores per chip clustered up to 1,024 chips

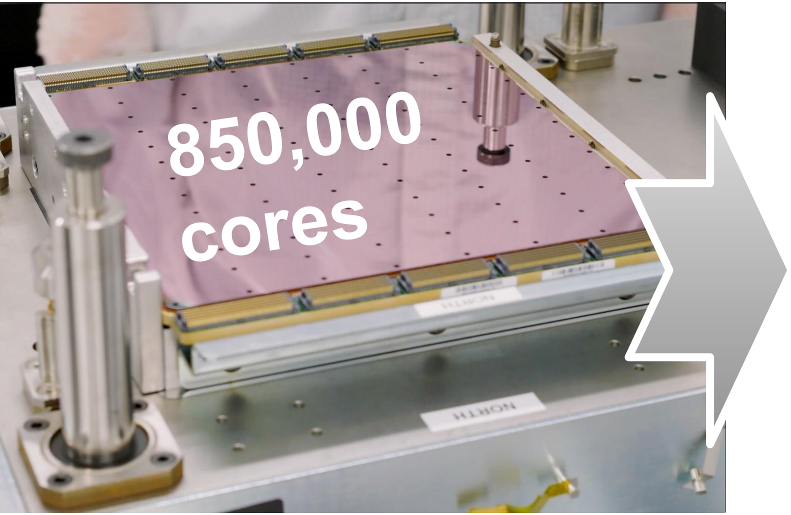




NVIDIA A100 GPU
6,912 CUDA cores

SambaNova RDU

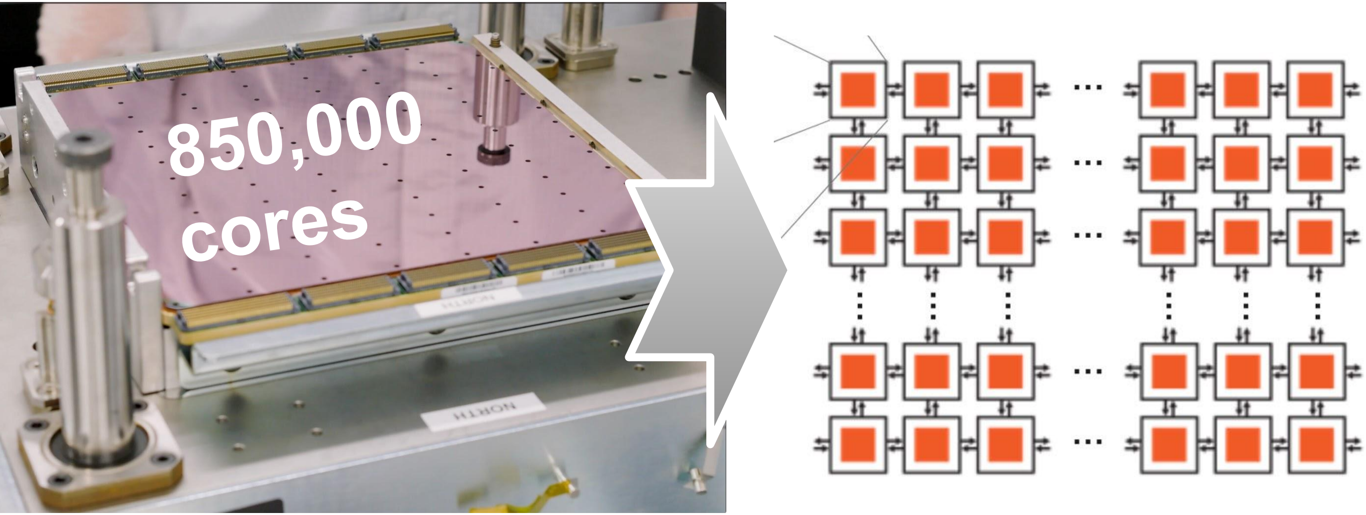
Cerebras Wafer-Scale Engine



850,000 cores

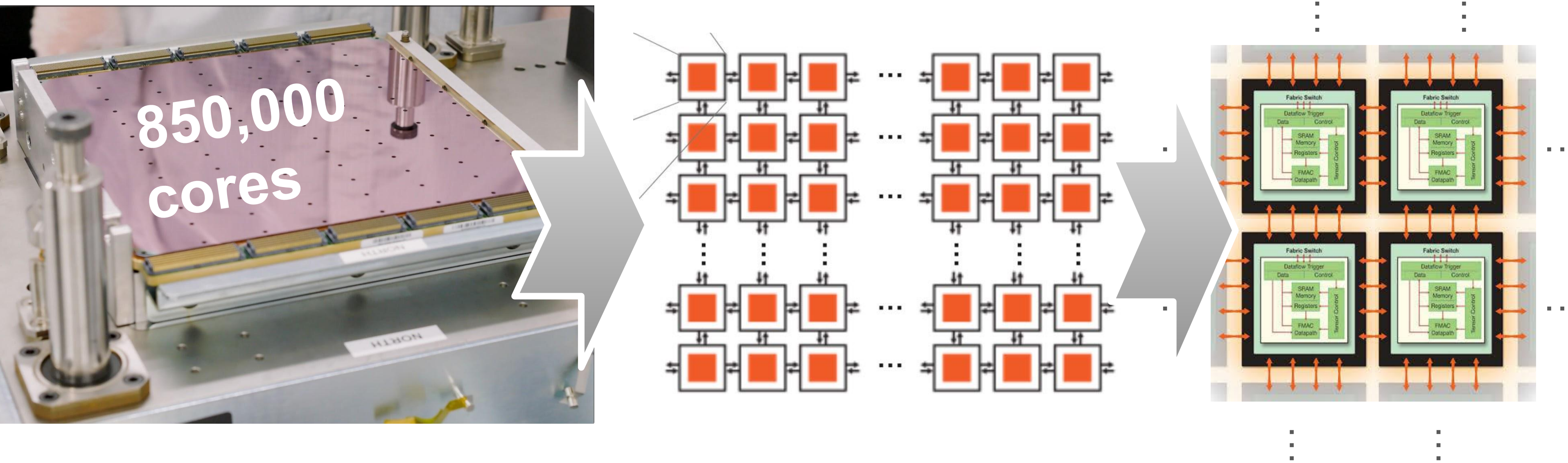
morenoma@umich.edu

Cerebras Wafer-Scale Engine



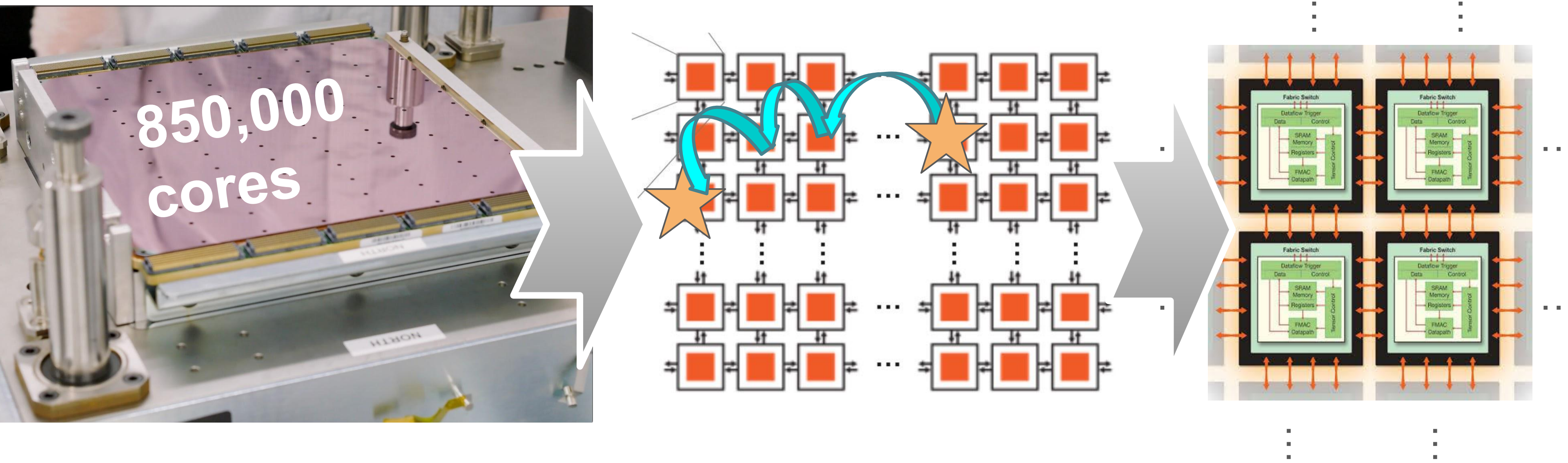
morenoma@umich.edu

Cerebras Wafer-Scale Engine



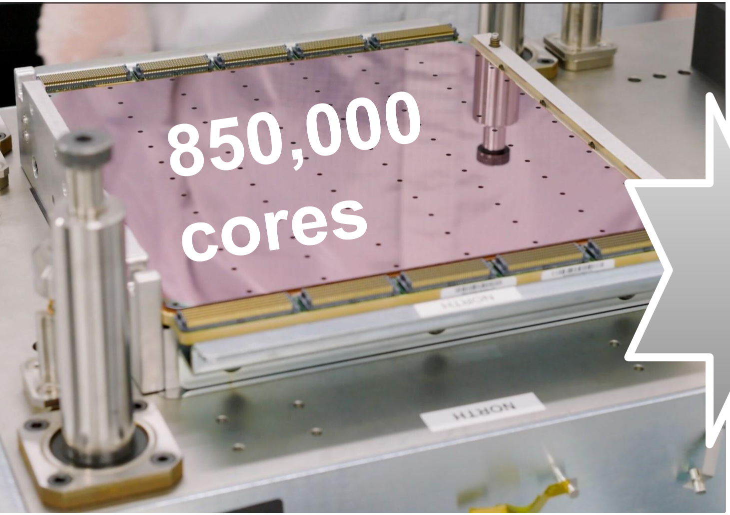
morenoma@umich.edu

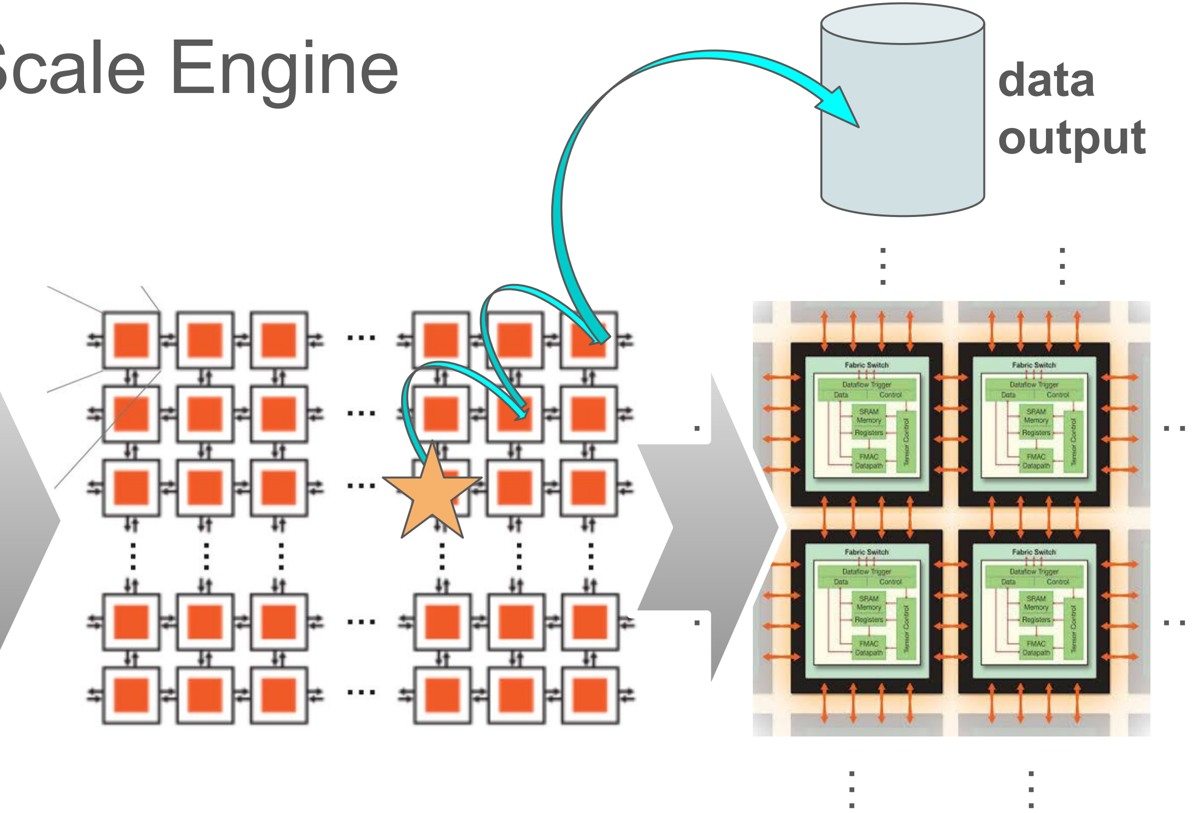
Cerebras Wafer-Scale Engine



morenoma@umich.edu

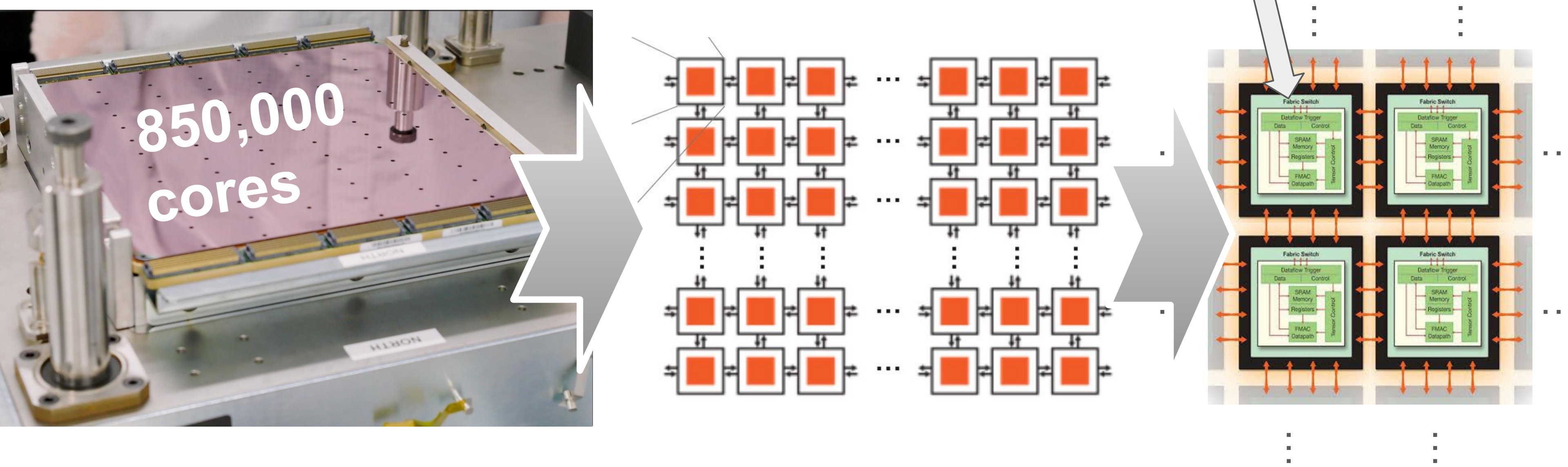
Cerebras Wafer-Scale Engine





morenoma@umich.edu

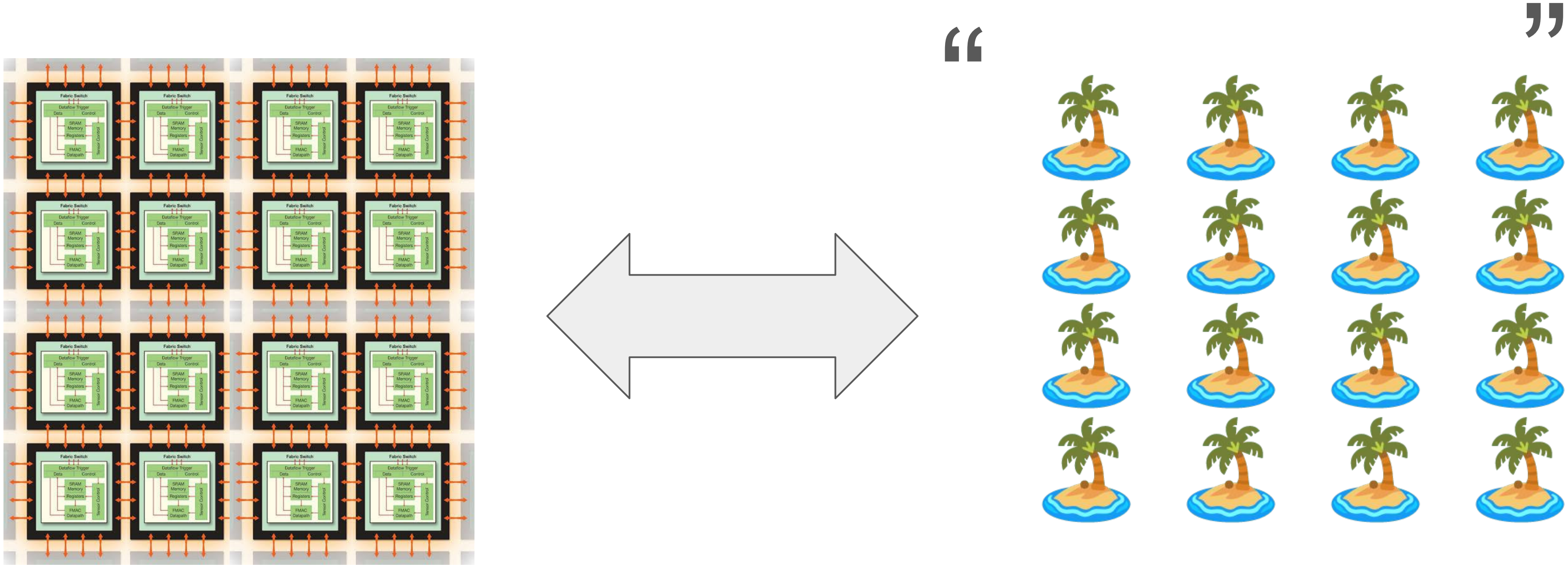
Cerebras Wafer-Scale Engine


850,000 cores

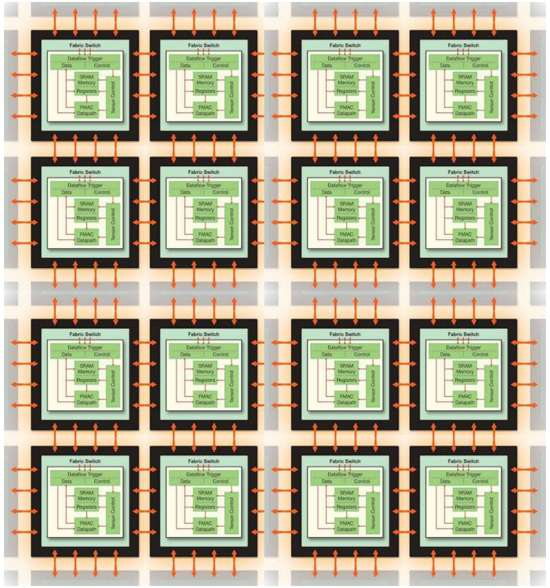
only ~48kb mem per core

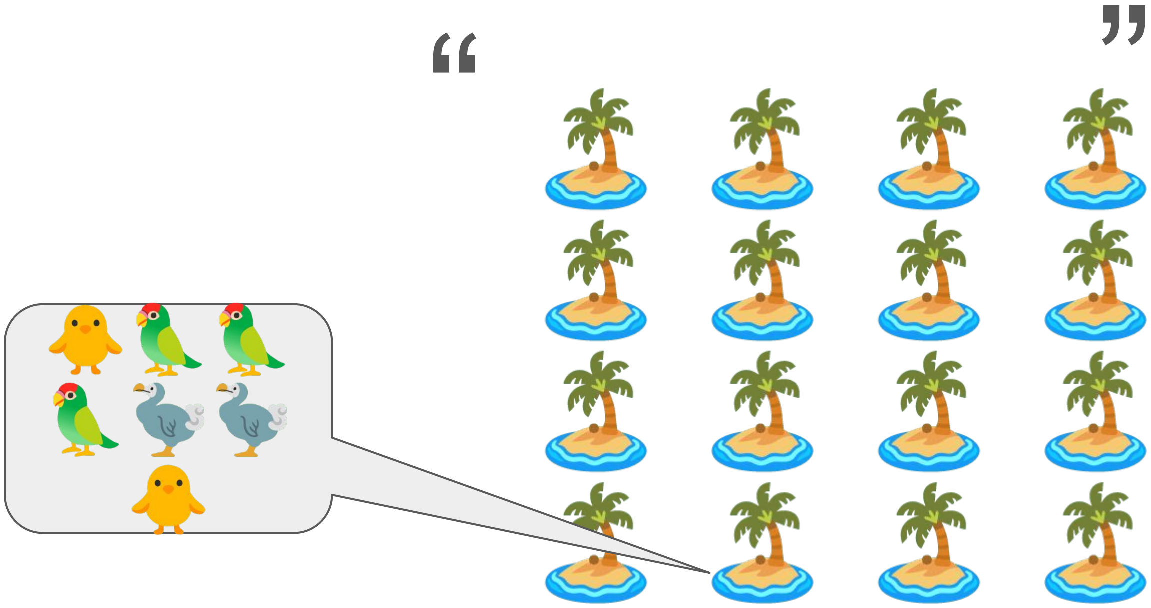
morenoma@umich.edu

Mapping Evolution Simulations onto WSE

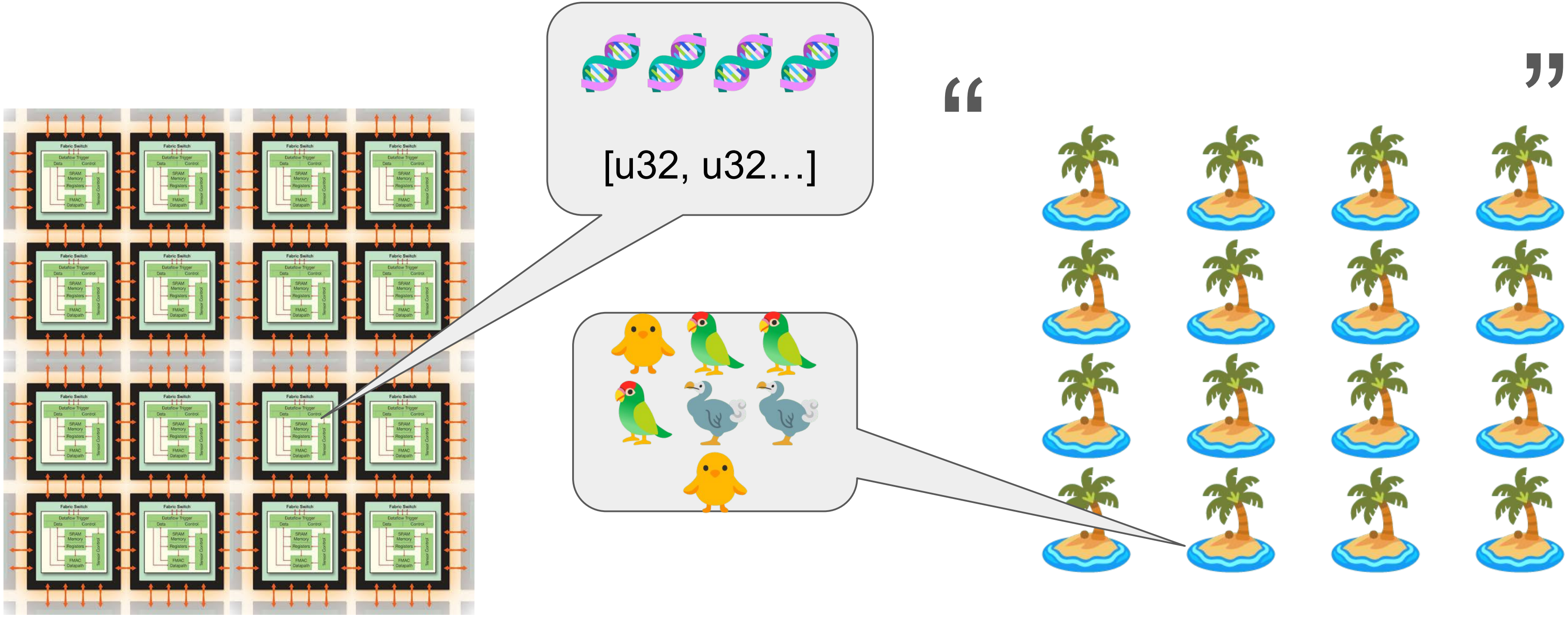


Mapping Evolution Simulations onto WSE



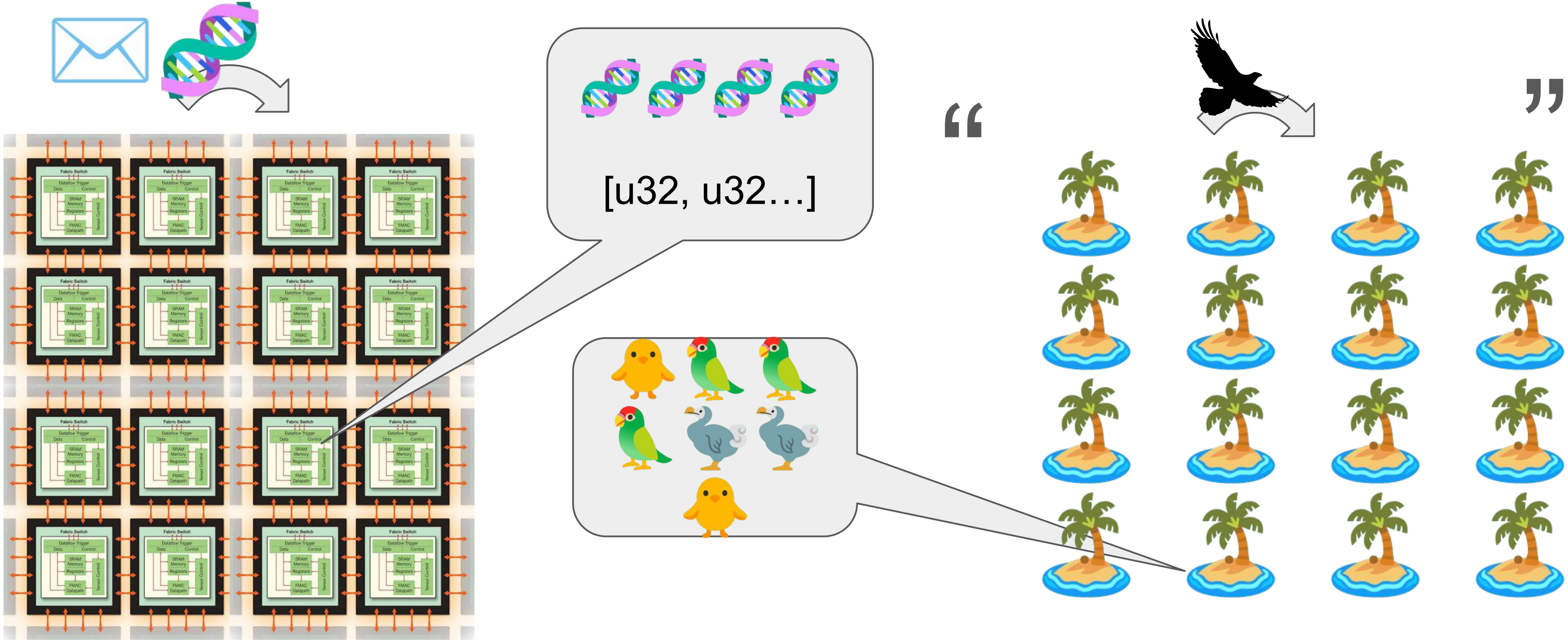


Mapping Evolution Simulations onto WSE



[u32, u32…]

Mapping Evolution Simulations onto WSE



[u32, u32…]

ByteBoost Workshop ‘24

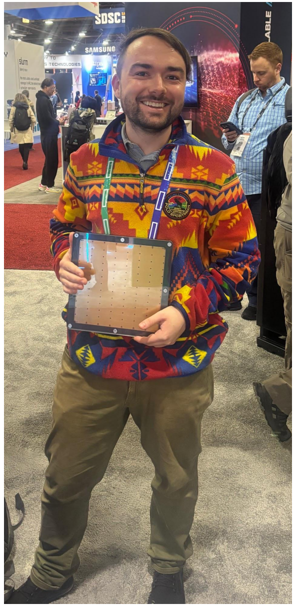
PSC

NEOCORTEX

ByteBoost Workshop ‘24

PSC

NEOCORTEX



ByteBoost Workshop '24

PSC

NEOCORTEX

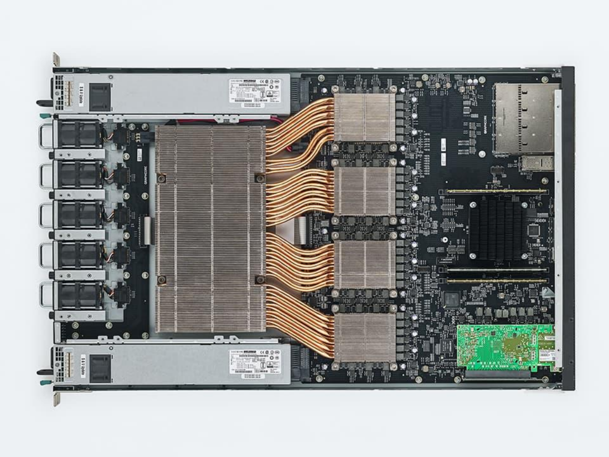
ByteBoost
Workshop ‘24

PSC

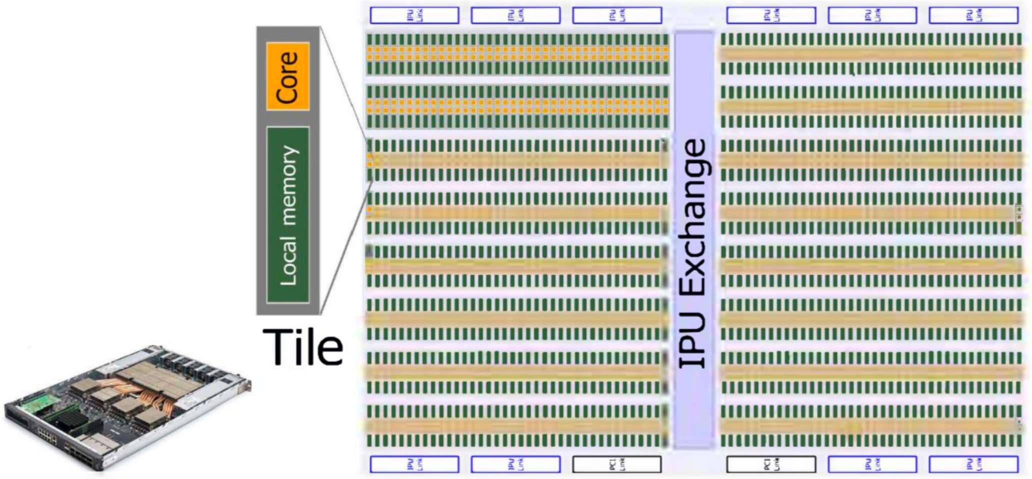
NEOCORTEX

fun facts!

GraphCore IPU — another AI/ML accelerator

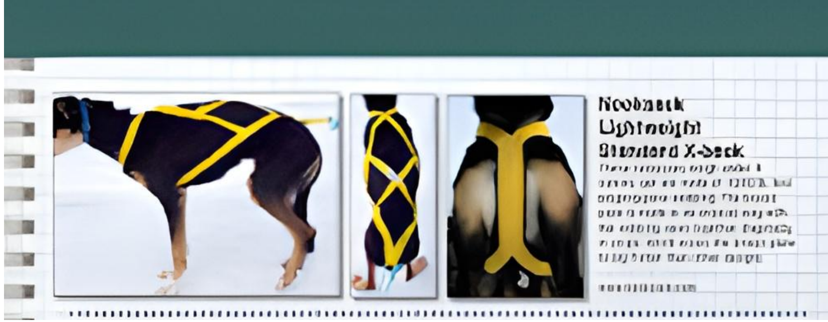


1,200 cores per chip
clustered up to 1,024 chips

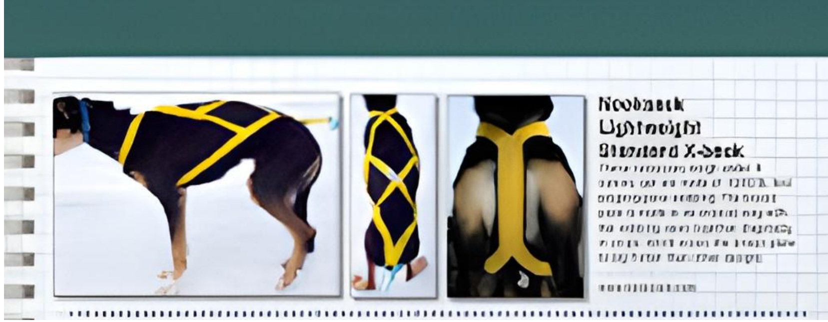


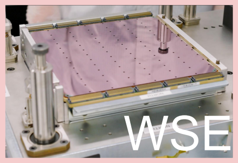
(a) *Graphcore IPU architecture and device*

Goal: develop and apply methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations

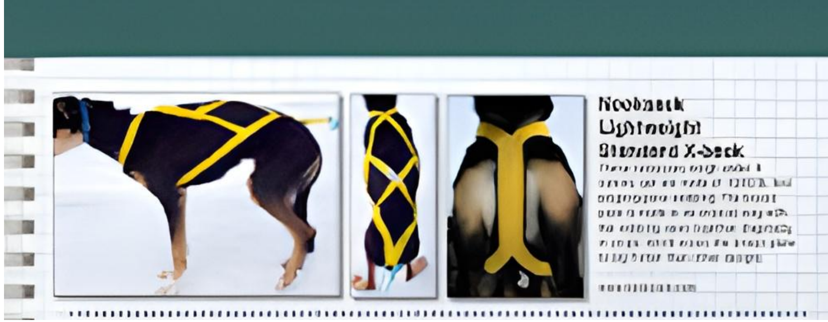


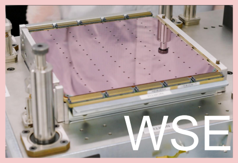
Goal: develop and apply methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations

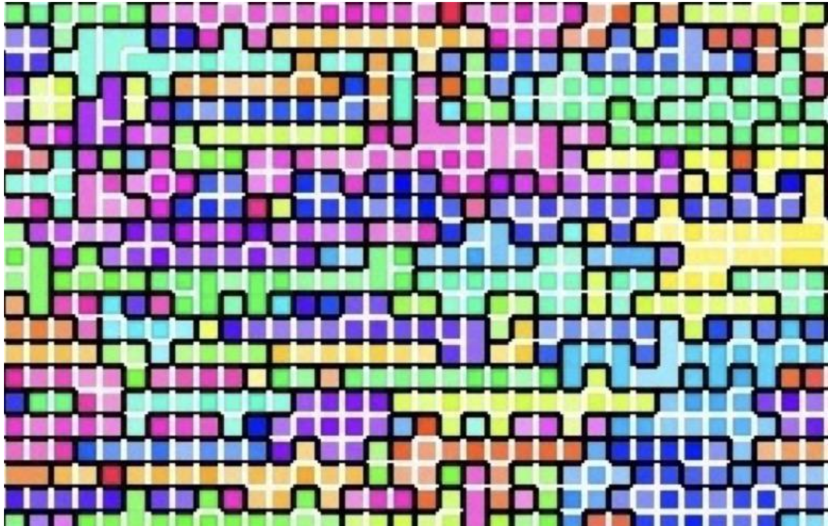




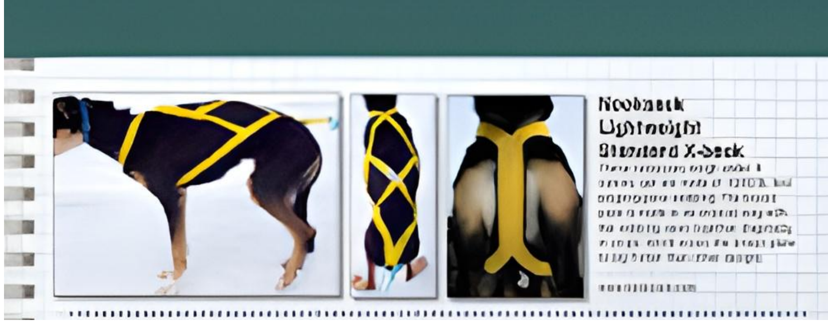
Goal: develop and apply methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations

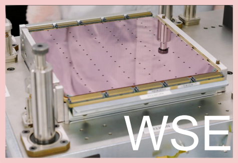


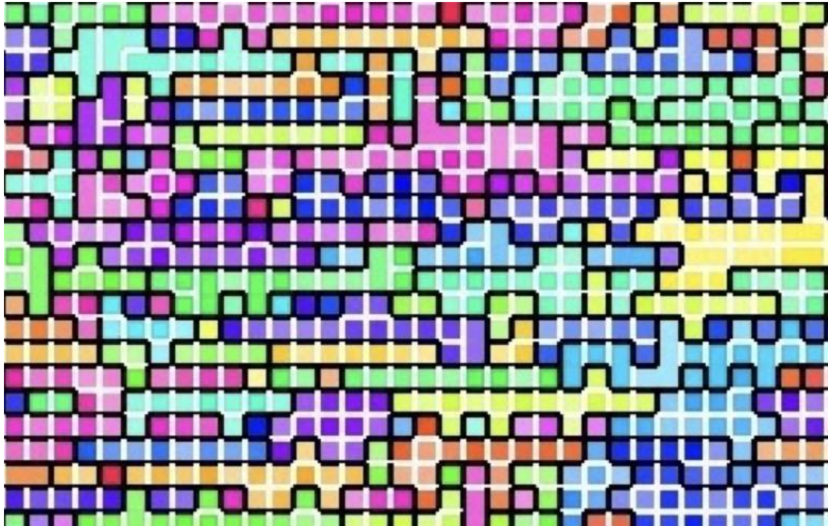




Goal: develop and apply methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations







Part 1:
Decentralized Approximate Phylogeny Tracking

Part 2:
Cerebras SDK Workflow on Wafer-Scale Cloud

Goal: develop and apply methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations

Part 1:
Hypermutator Dynamics Experiments on the Wafer-scale Engine

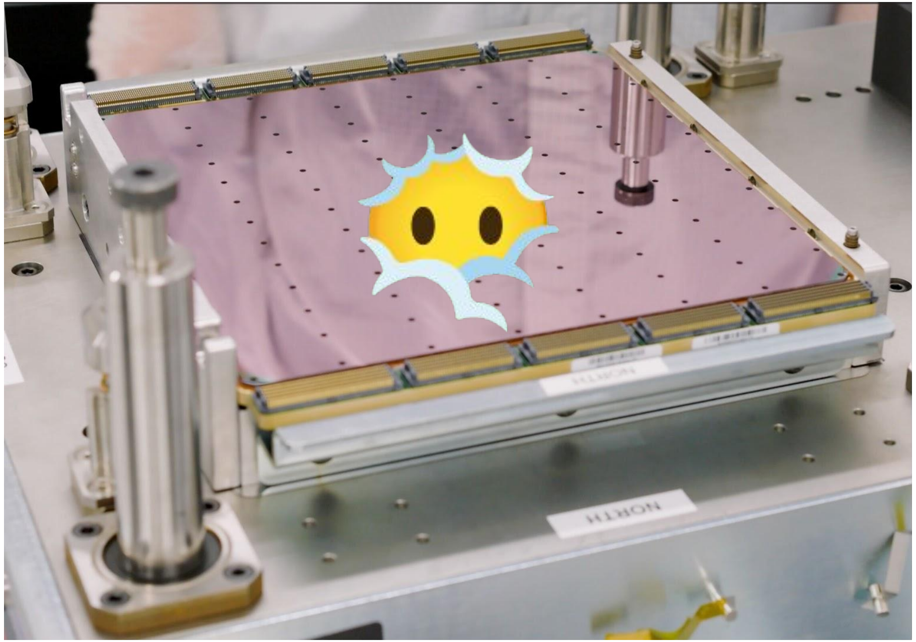
Part 2:
Decentralized Phylogeny Tracking

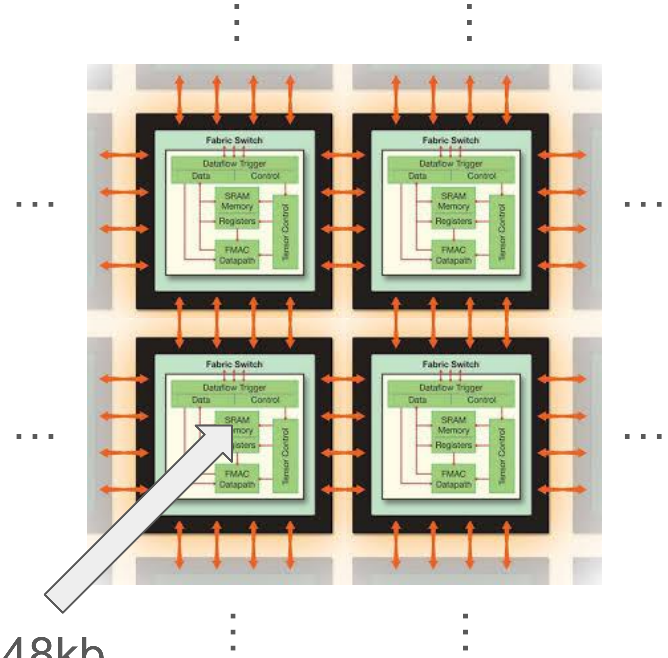
Part I:

Decentralized Approximate Phylogeny Tracking

morenoma@umich.edu

Cerebras Wafer-Scale Engine

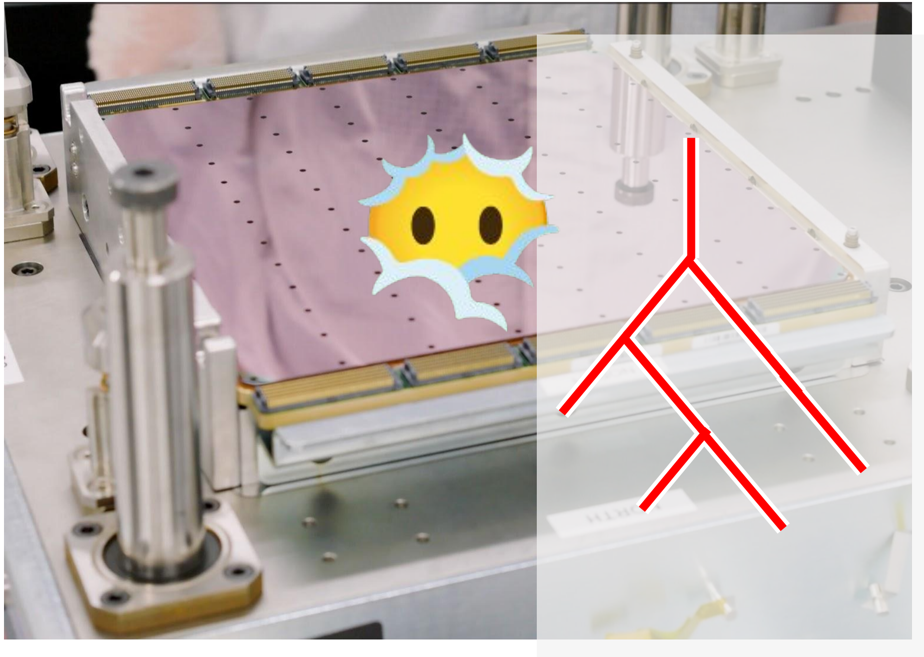




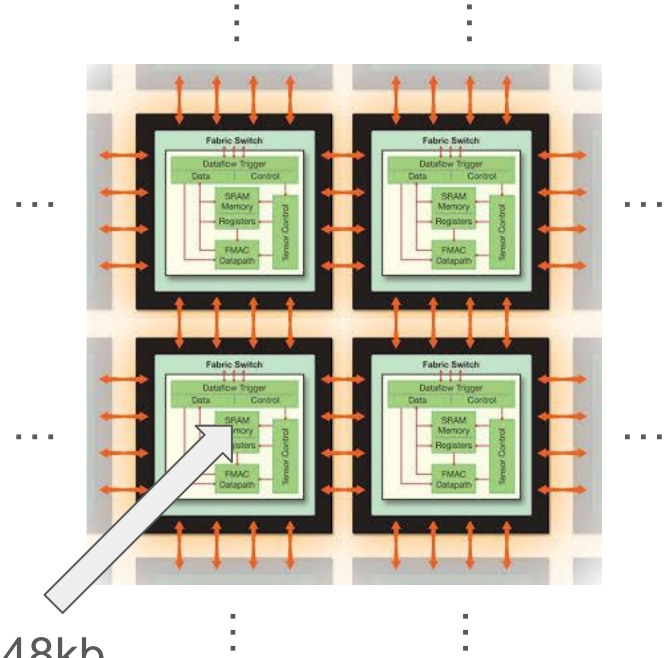
48kb
mem

morenoma@umich.edu

Cerebras Wafer-Scale Engine



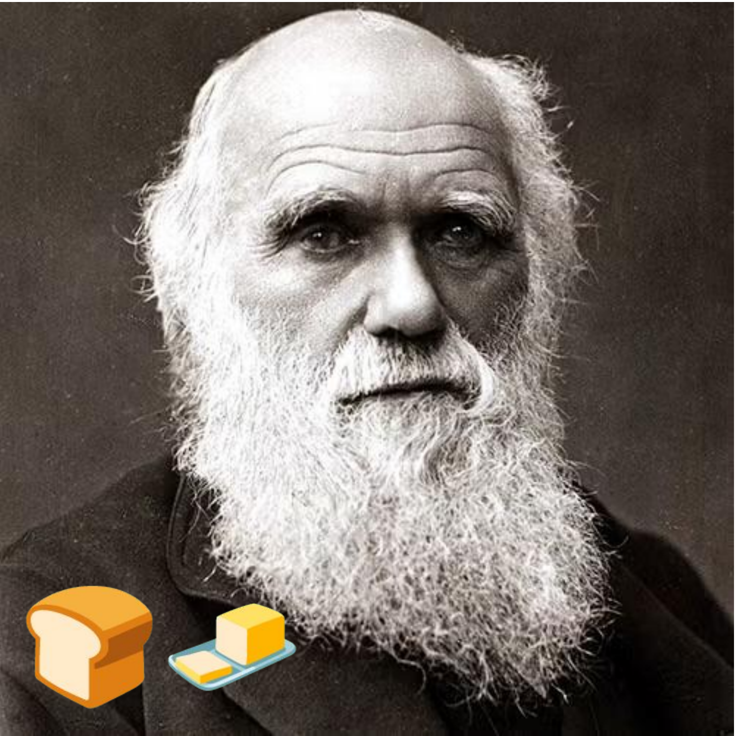
phylogeny

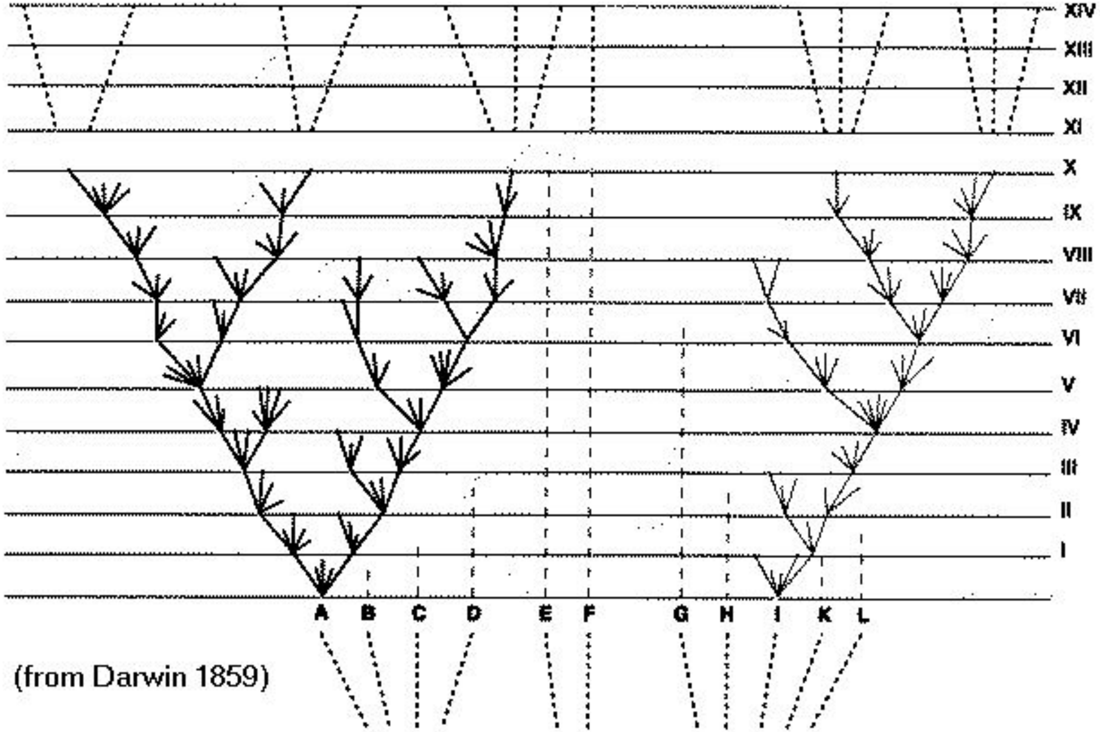


48kb
mem

📧 morenoma@umich.edu

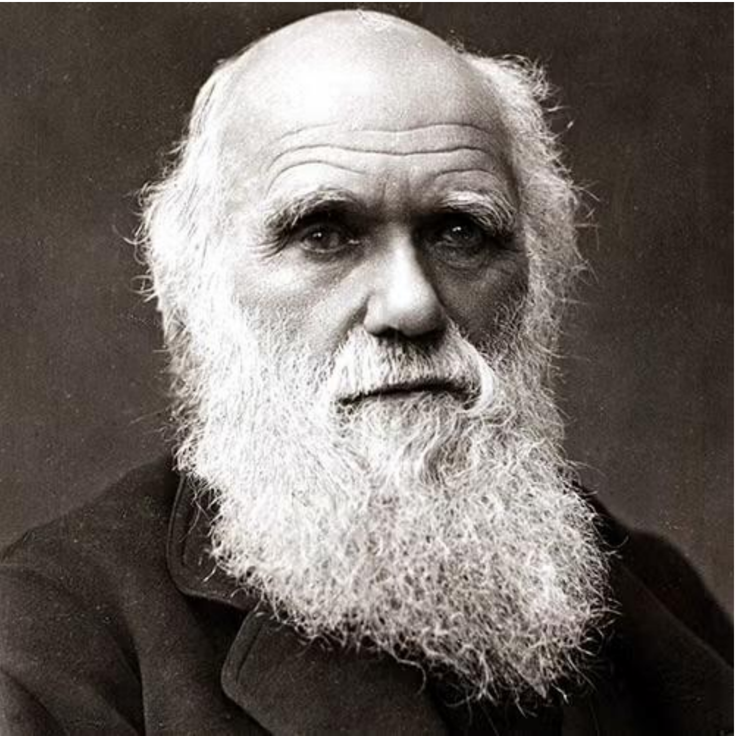
What is a phylogeny and what can it tell you?





(from Darwin 1859)

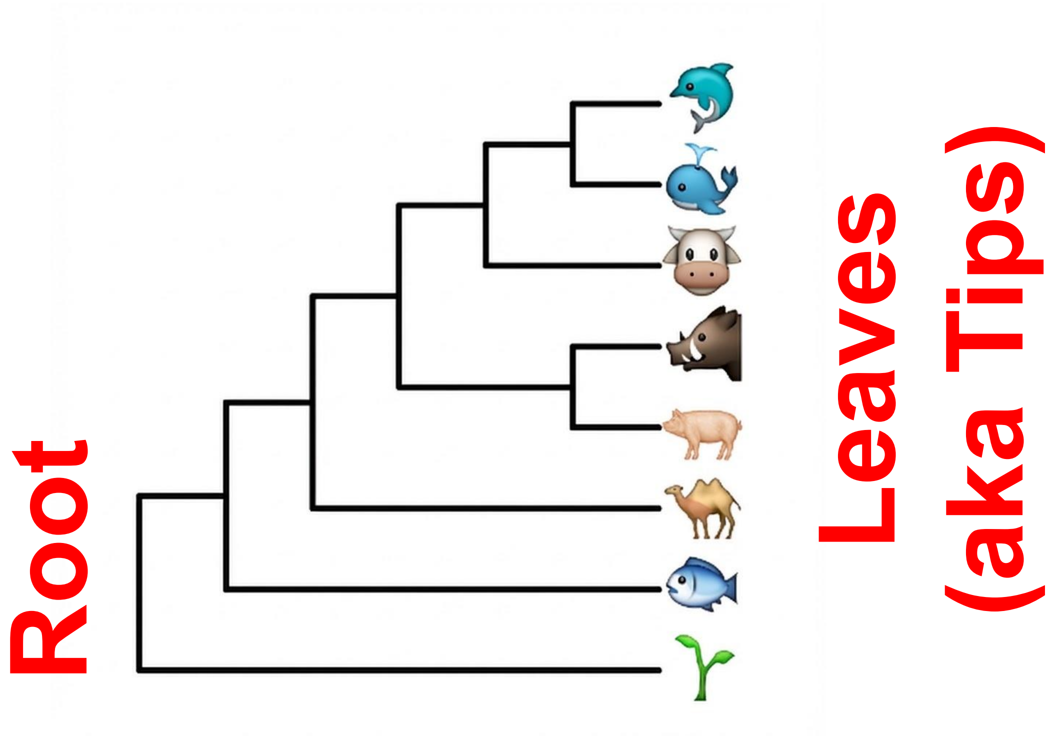
What is a phylogeny and what can it tell you?



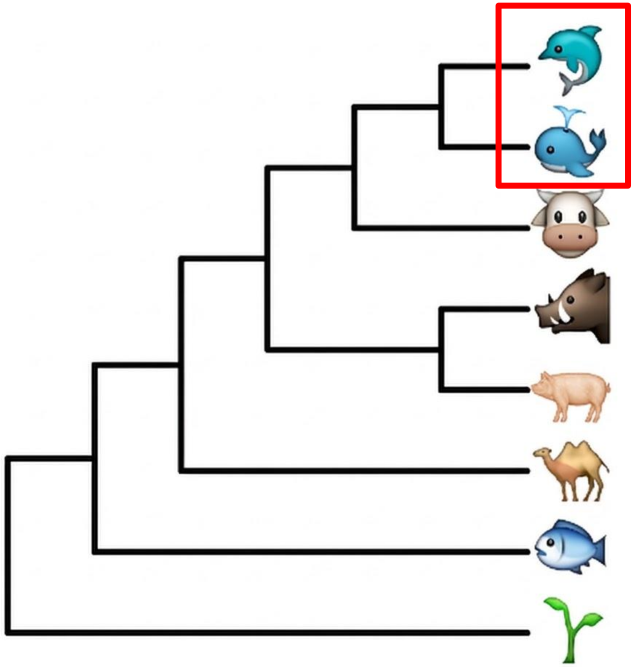
Simulation Phylogeny Data:

1. history of evolutionary events
2. insight into evolutionary dynamics
3. “ground truth” data to test/develop bioinformatics tools

What is a phylogeny and what can it tell you?

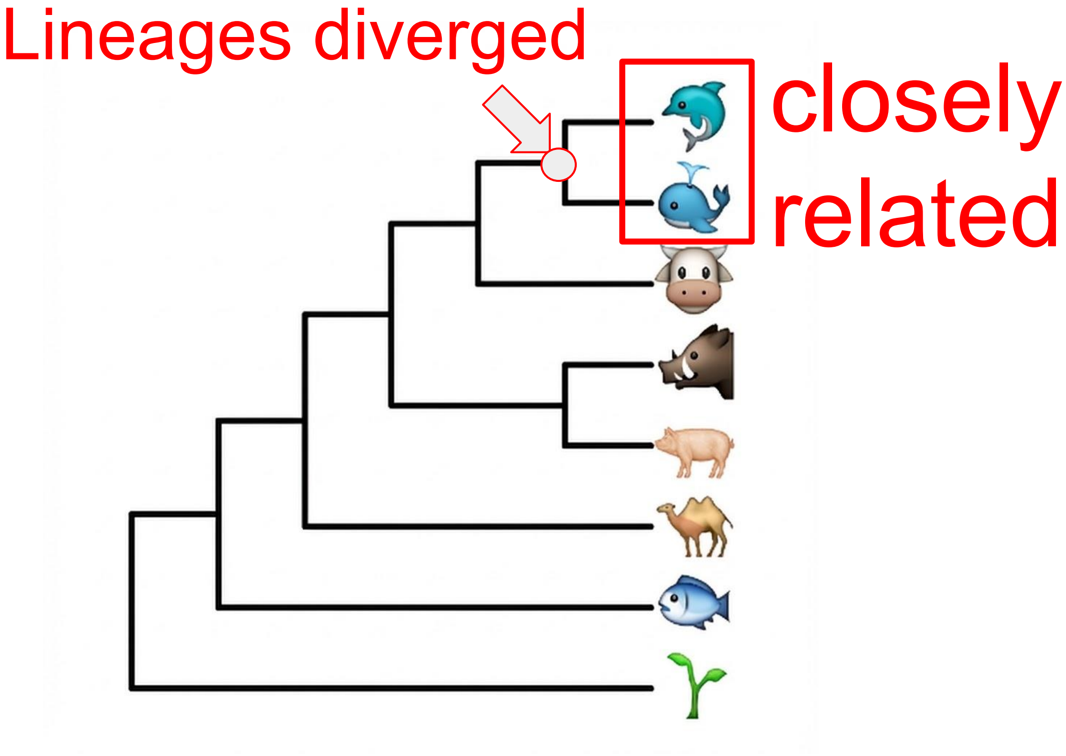


What is a phylogeny and what can it tell you?

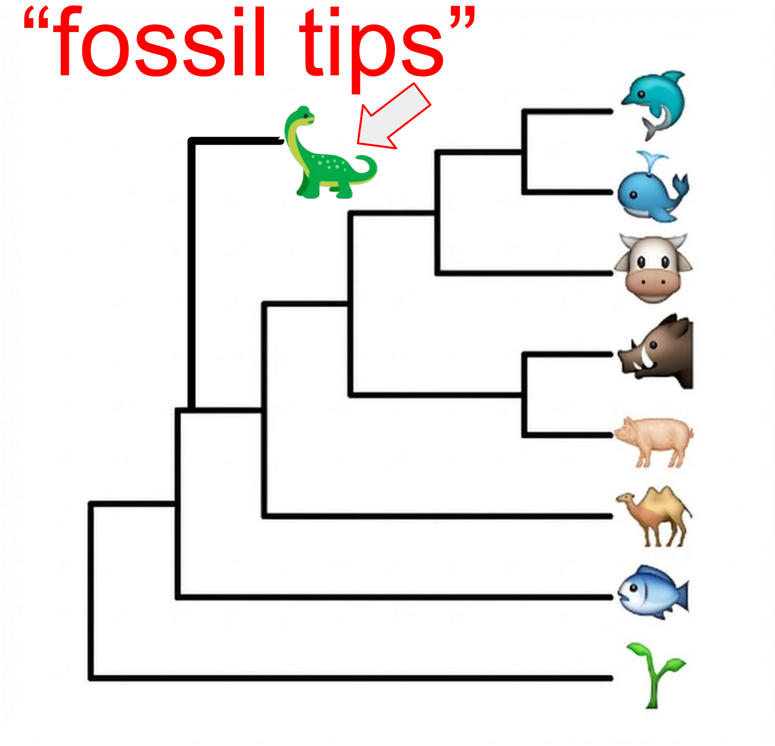


closely
related

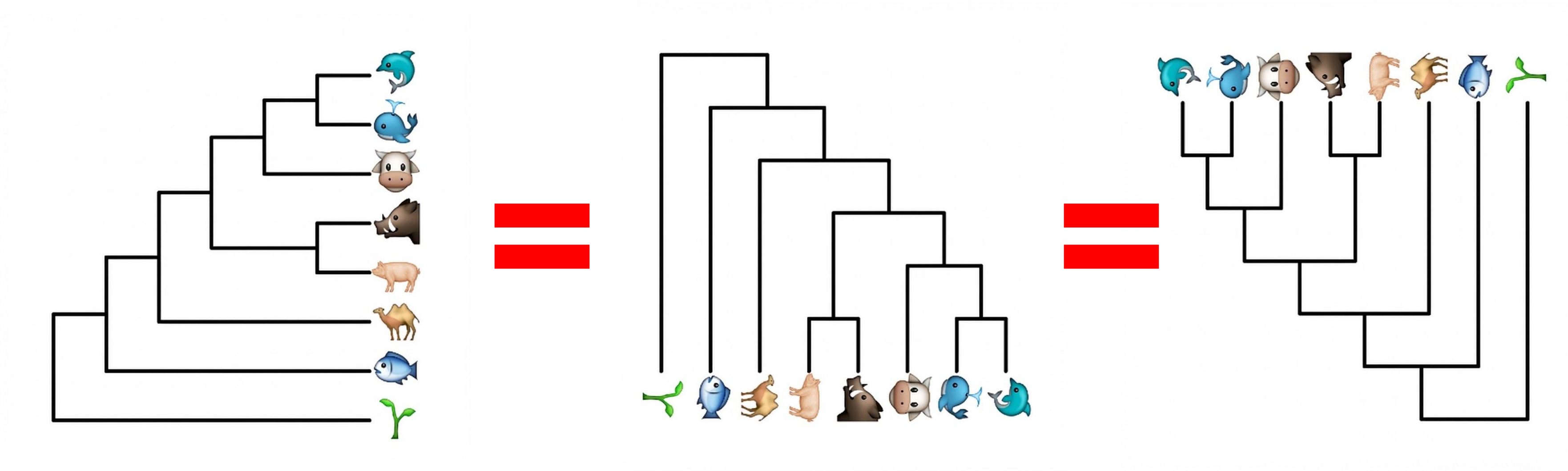
What is a phylogeny and what can it tell you?



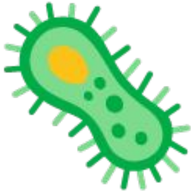
What is a phylogeny and what can it tell you?

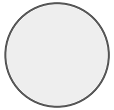


What is a phylogeny and what can it tell you?



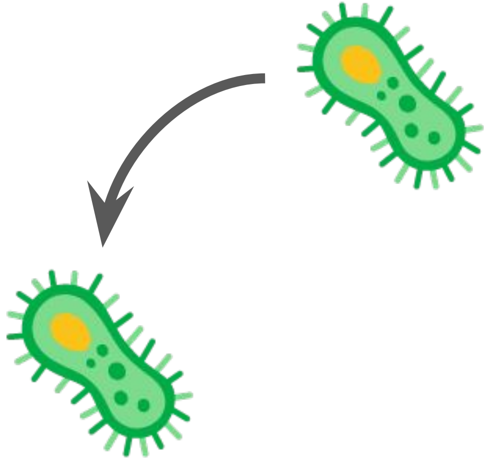
Asexual Phylogeny

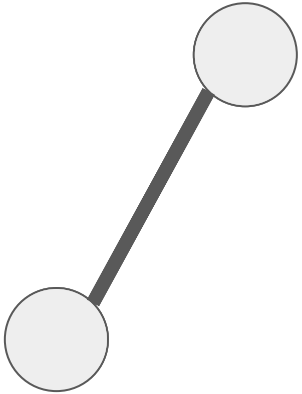




vertex:
“taxonomic unit”

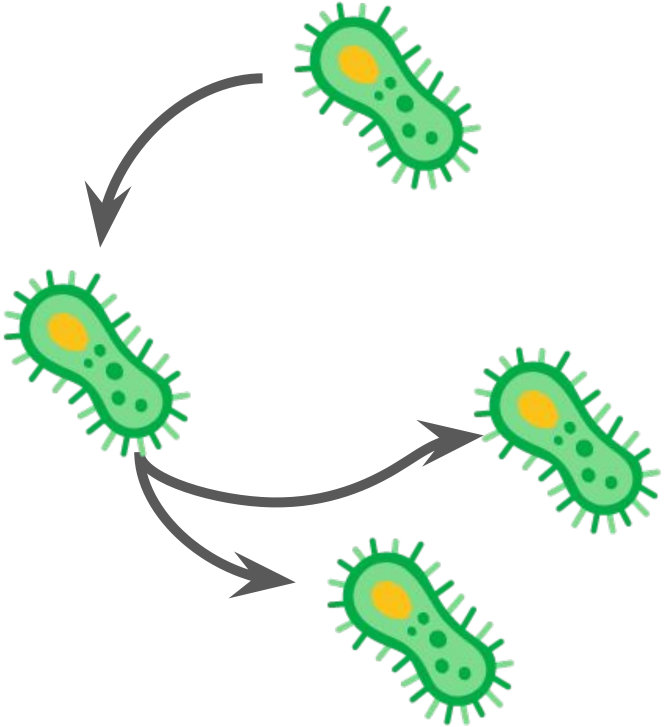
Asexual Phylogeny

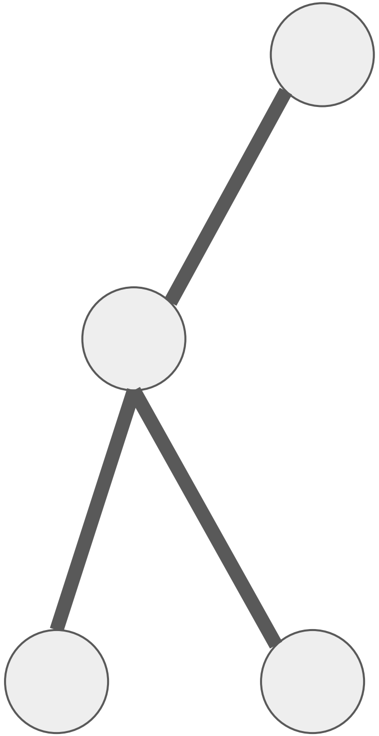




vertex:
“taxonomic unit”

Asexual Phylogeny

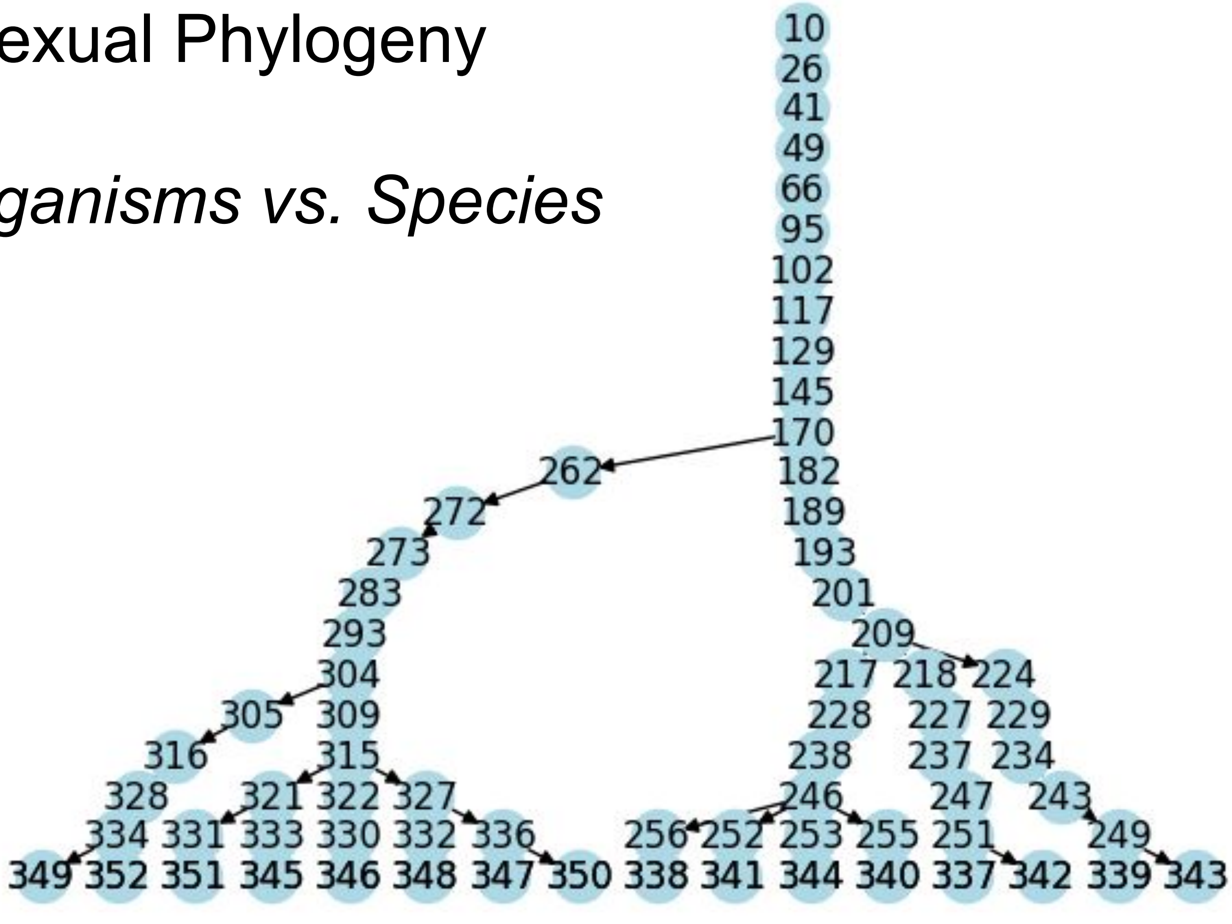




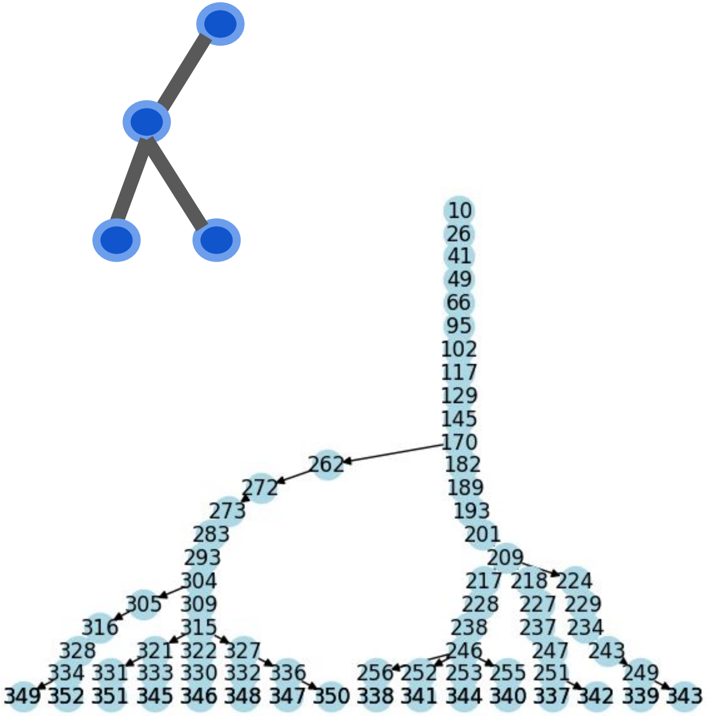
vertex:
“taxonomic unit”

Asexual Phylogeny

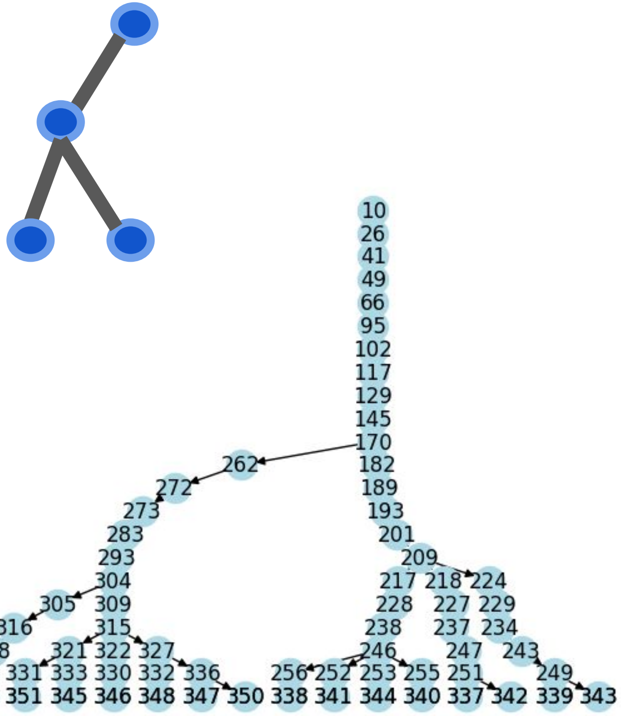
Organisms vs. Species



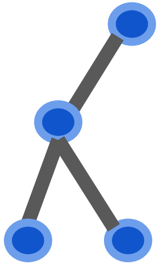
Sexual vs asexual phylogenies

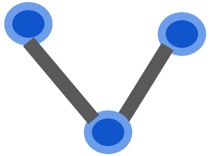


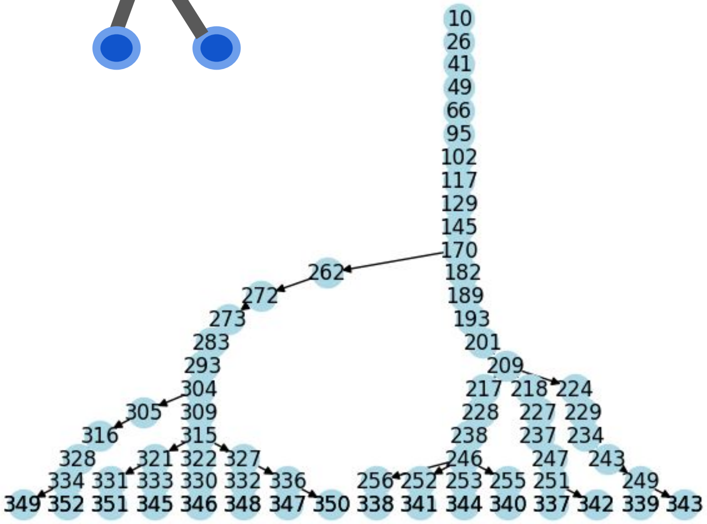
Sexual vs asexual phylogenies



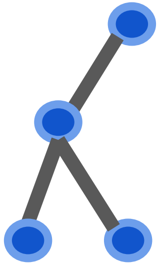
Sexual vs asexual phylogenies

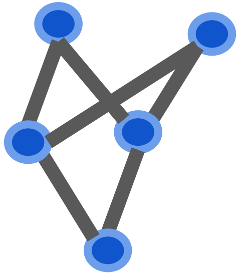


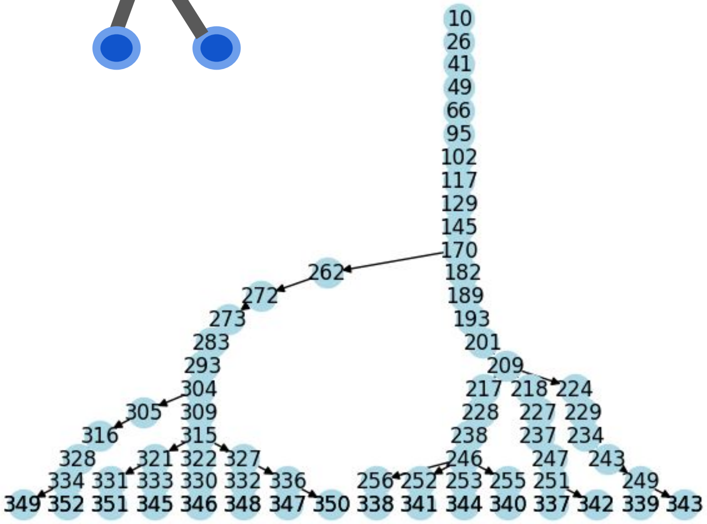




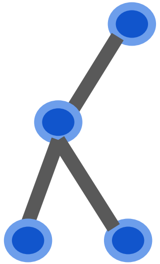
Sexual vs asexual phylogenies

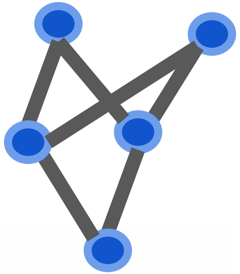


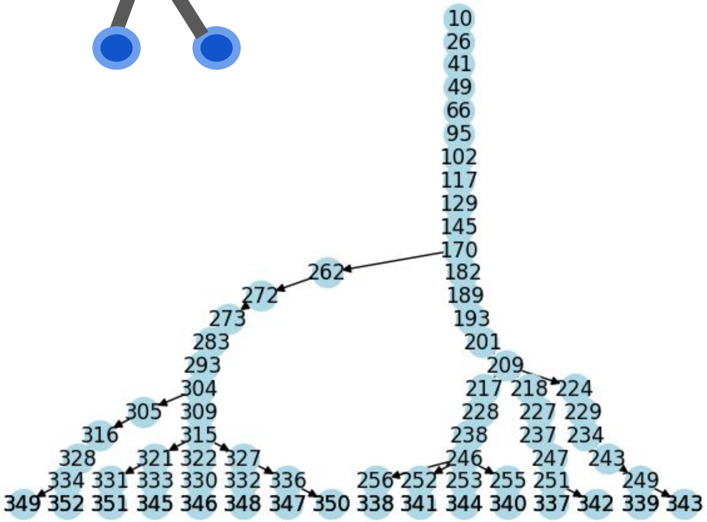


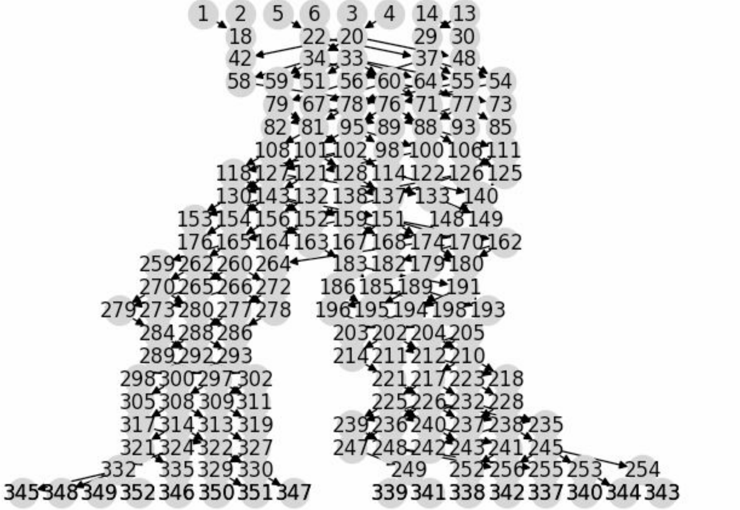


Sexual vs asexual phylogenies

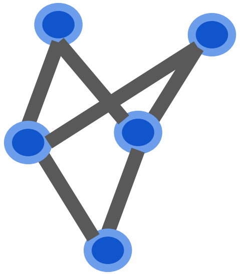




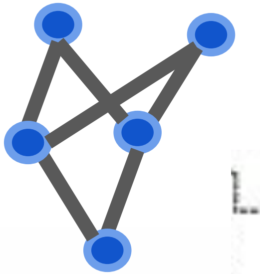


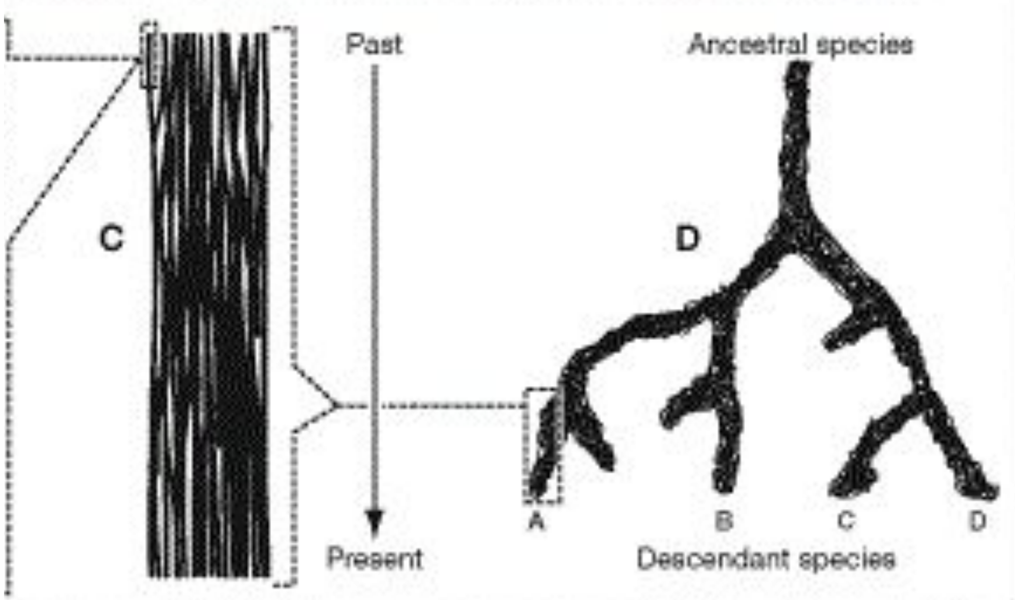


Sexual vs asexual phylogenies



Sexual vs asexual phylogenies

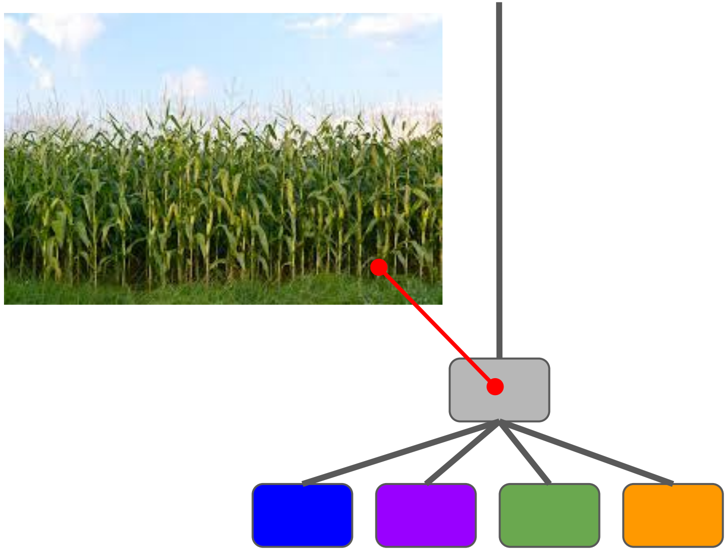




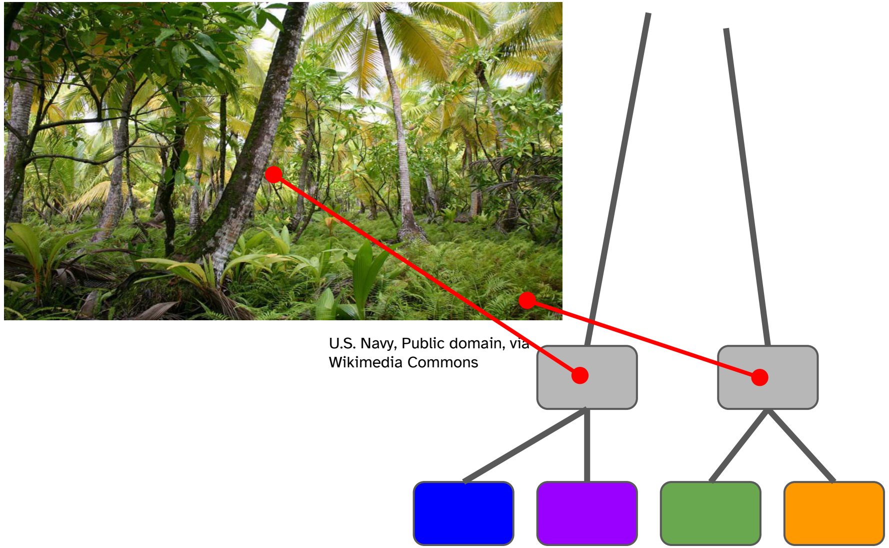
Baum and Offner, 2008

What is a phylogeny and what can it tell you?

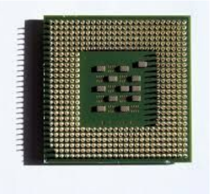
recent selective sweep



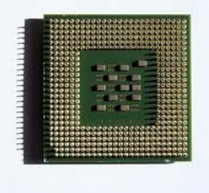
long term coexistence



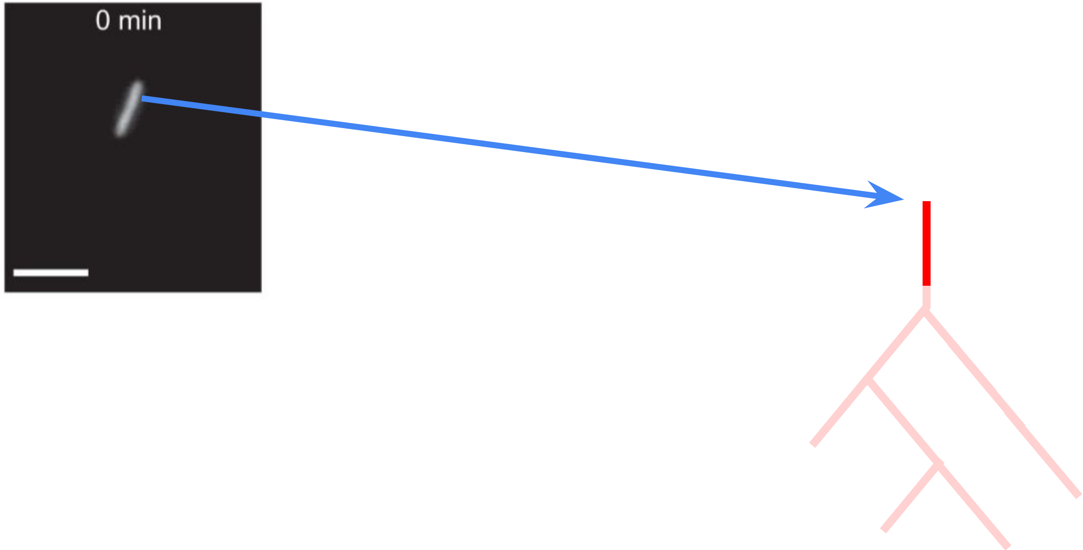
U.S. Navy, Public domain, via Wikimedia Commons



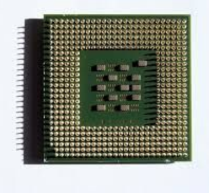
Perfect Tracking:
Complete Observability



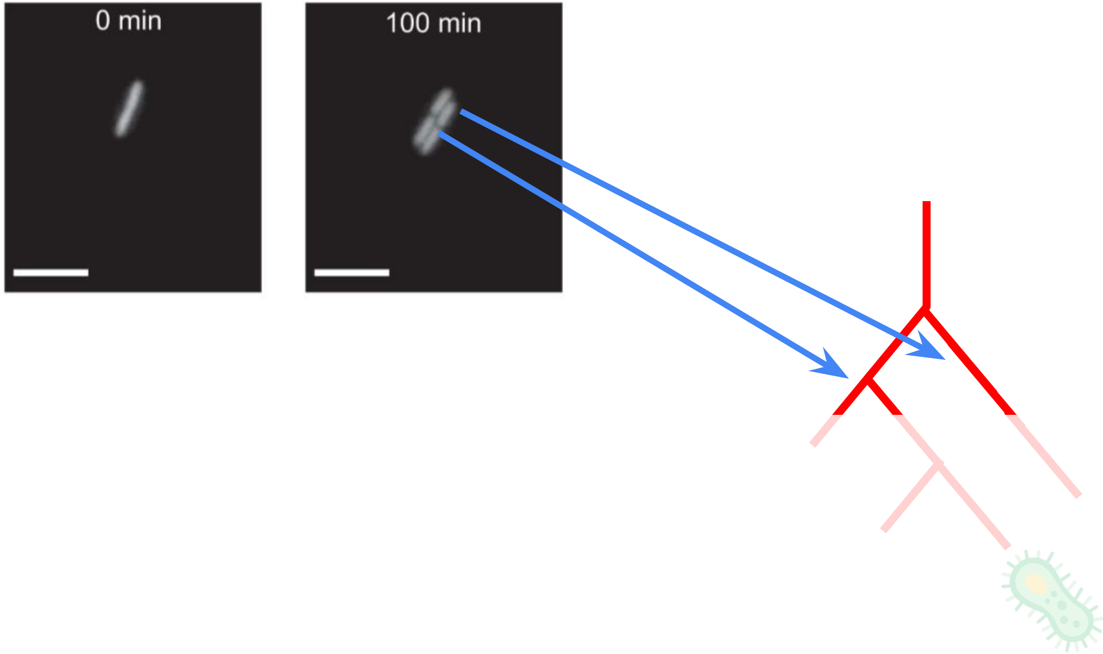
Perfect Tracking:
Complete Observability



(Nozoe et al, 2017)

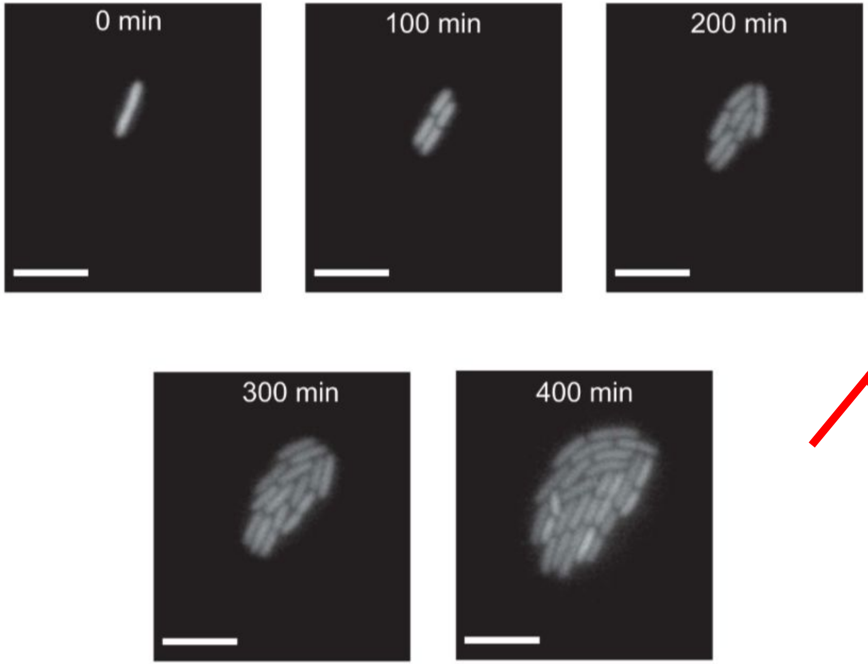


Perfect Tracking:
Complete Observability

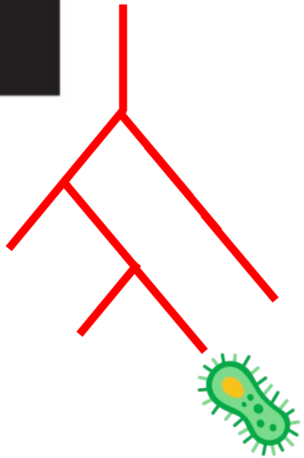


(Nozoe et al, 2017)

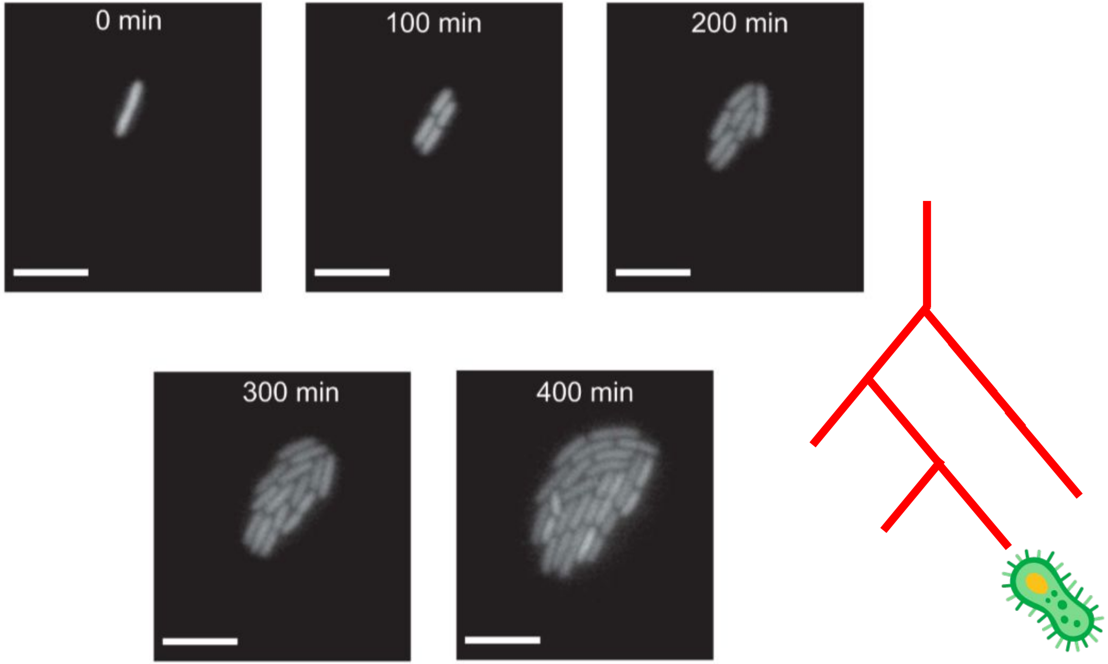
Perfect Tracking:
Complete Observability



(Nozoe et al, 2017)

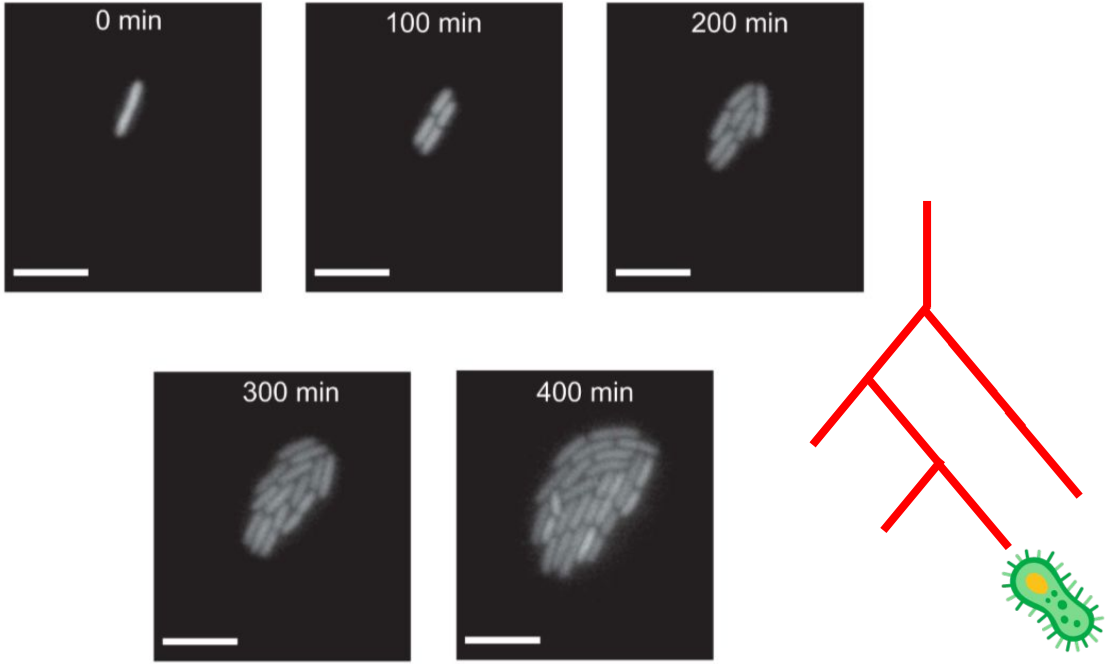


Perfect Tracking:
Complete Observability



(Nozoe et al, 2017)

Perfect Tracking: Complete Observability

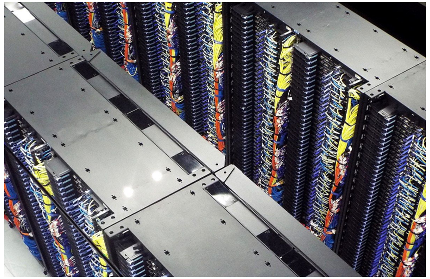


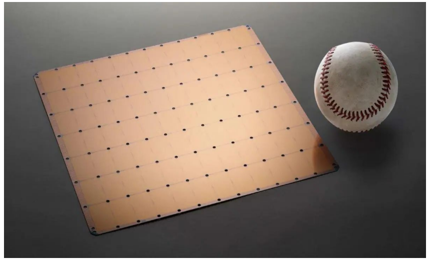
(Nozoe et al, 2017)

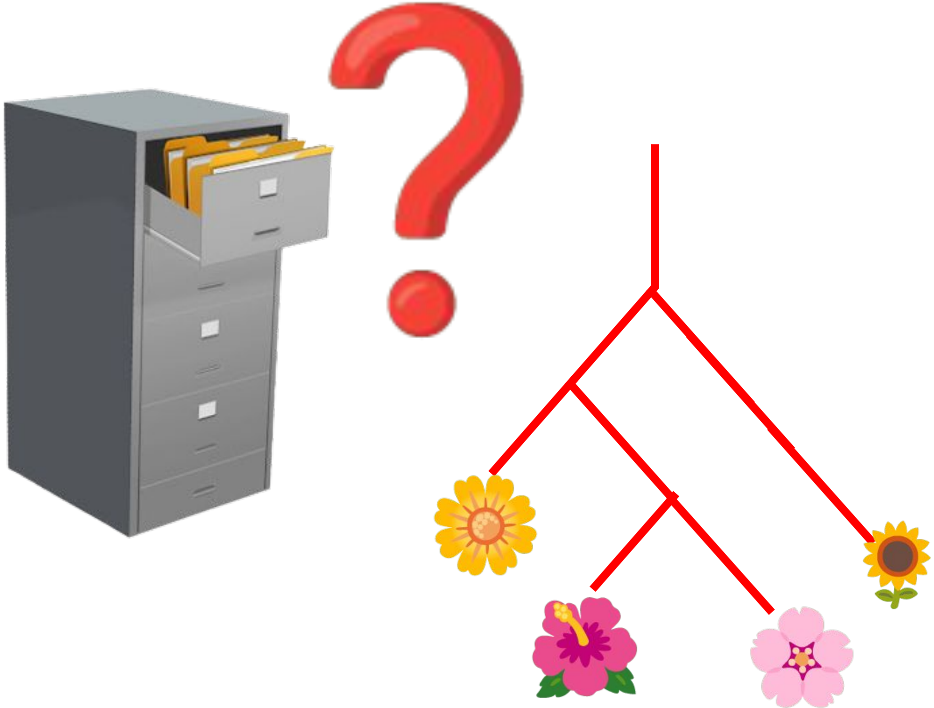
Costs of Perfect Tracking

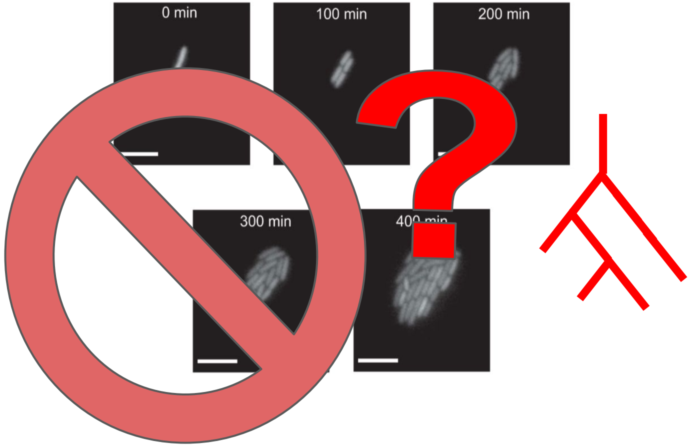
- Memory, Storage, & Bandwidth
  - e.g., 8PB vs 30GB
- Runtime Extinction Tracking Costs
- Hardware Crashout & Corruption
  - perfect tracking is sensitive to data loss
  - at exascale, >1 node fail/hour

Highly-distributed, Many-core Computation

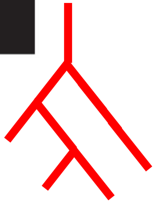




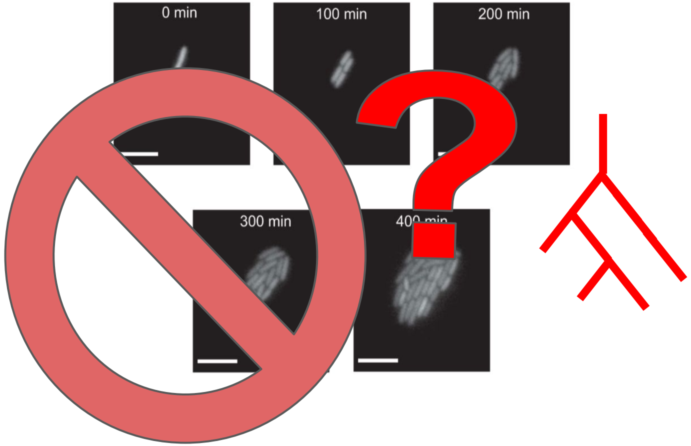


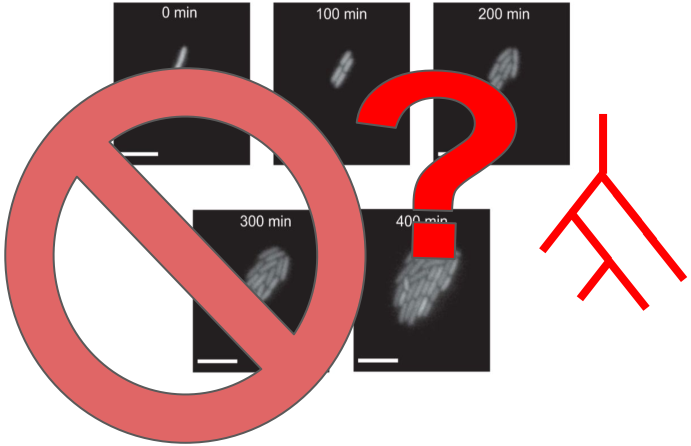


🐦@MorenoMatthewA 🐘 @mas.to

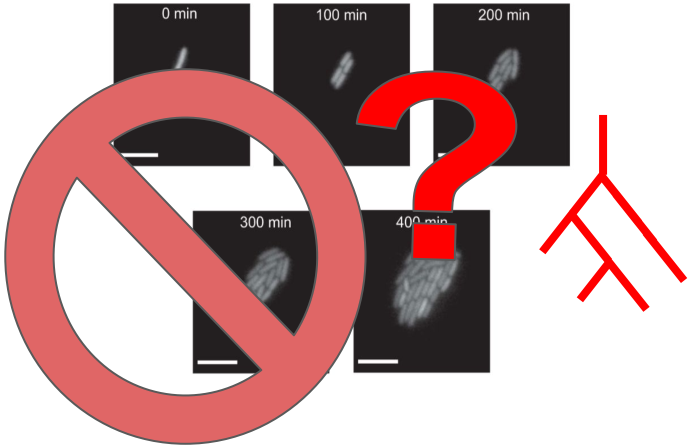


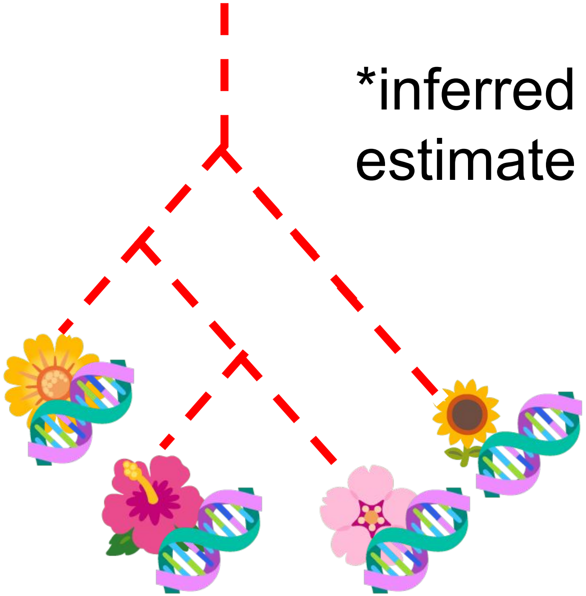
🐦@MorenoMatthewA 🐘@mas.to


0 min
100 min
200 min
300 min
400 min



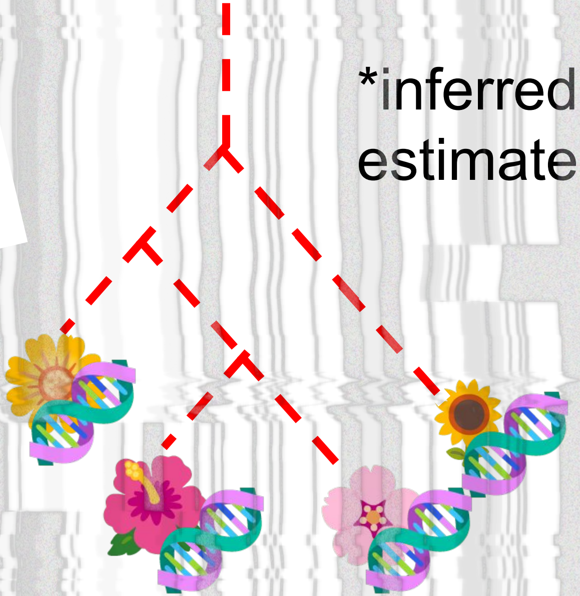
🐦@MorenoMatthewA 🐘 @mas.to





*inferred
estimate

How to design 🧬 to facilitate reconstruction?

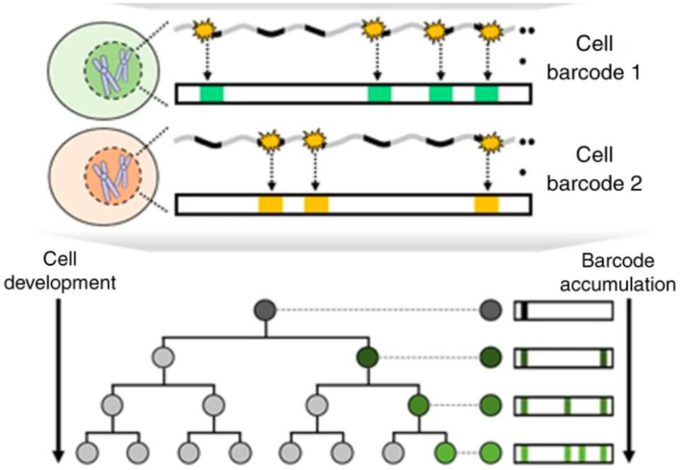


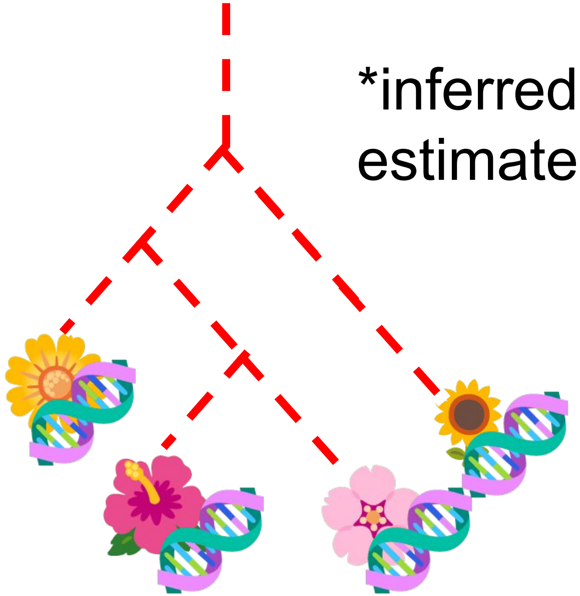
*inferred estimate

🐦@MorenoMatthewA 🐘 @mas.to

Hwang et al., 2019

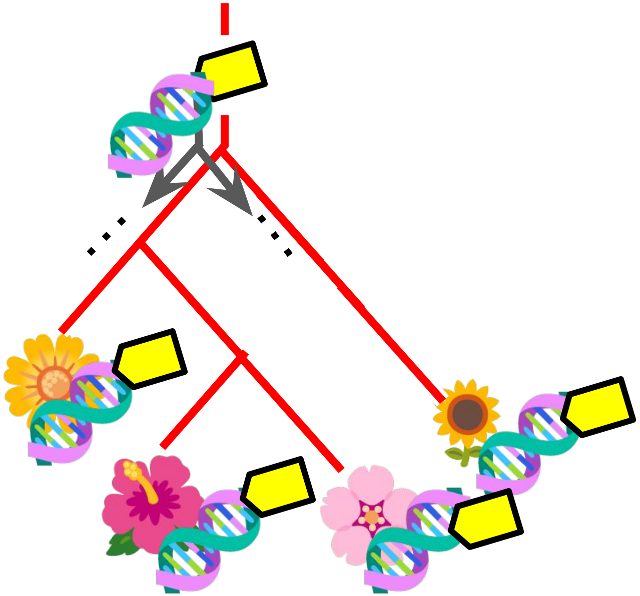
Crispr-cas9 Barcoding





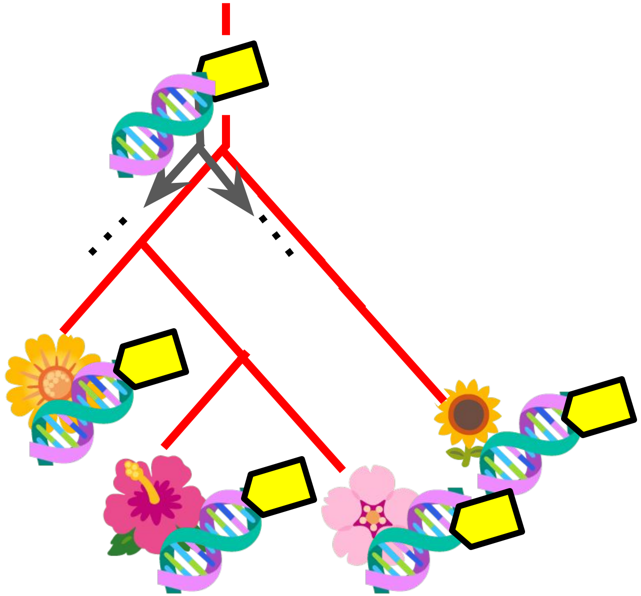
*inferred estimate

⬟ = “hstrat” annotation



⬟ = “hstrat” annotation

- small (~96 bits)
- fast (< 500ns)
- tunably accurate



evolve

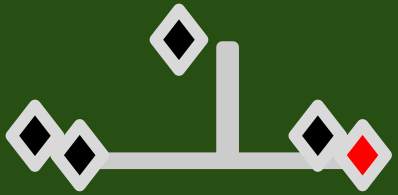
under-the-hood “hstrat” algorithm

evolve

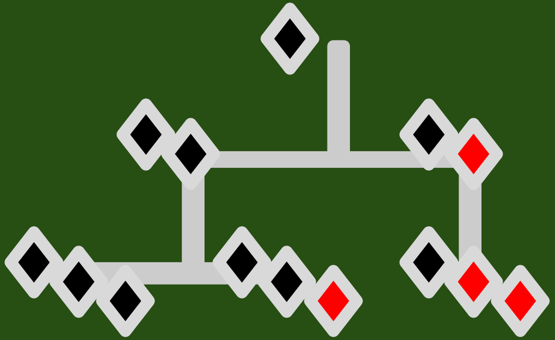
under-the-hood “hstrat” algorithm (✂️ for time)

evolve

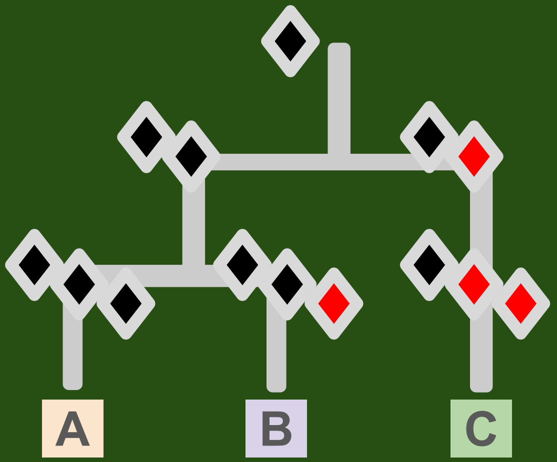
evolve



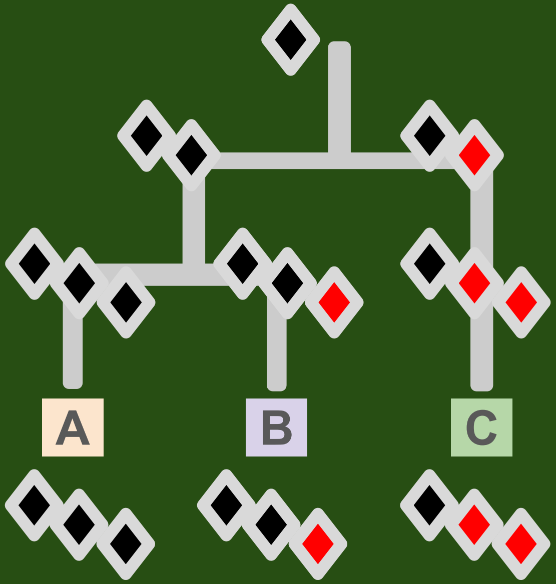
evolve



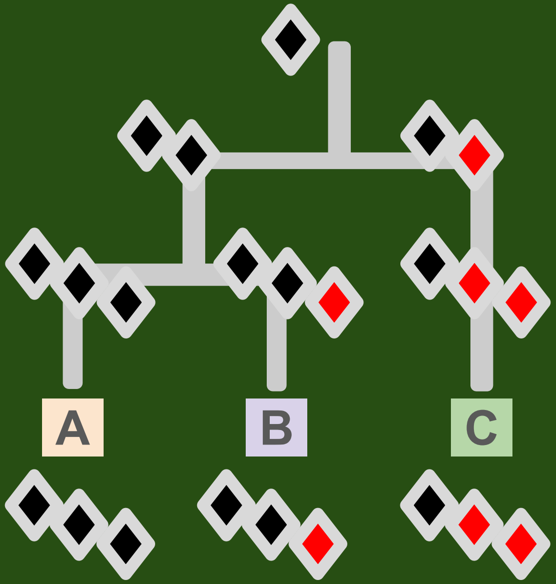
evolve



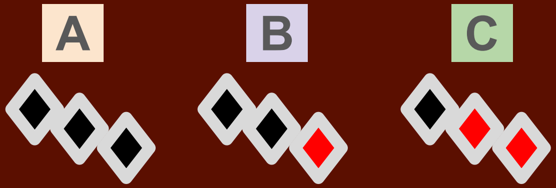
evolve



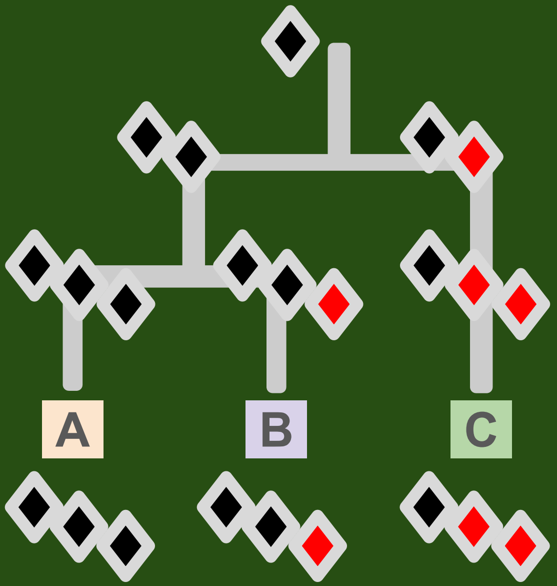
evolve



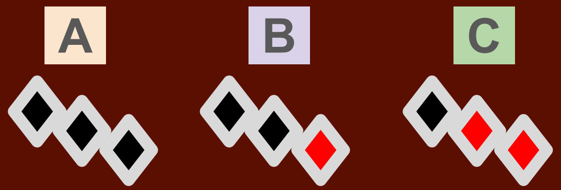
end-state



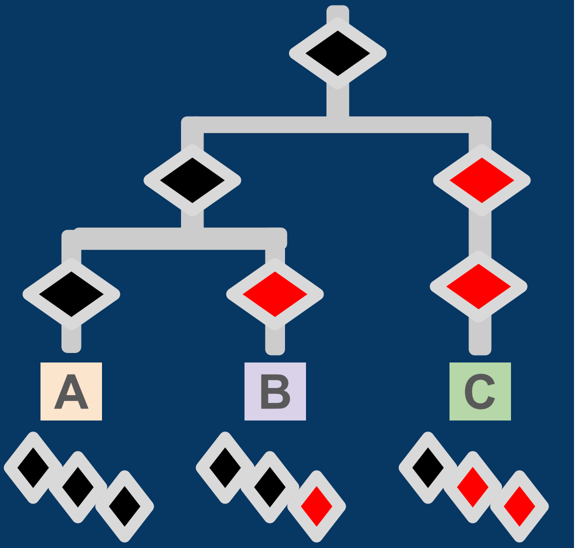
evolve

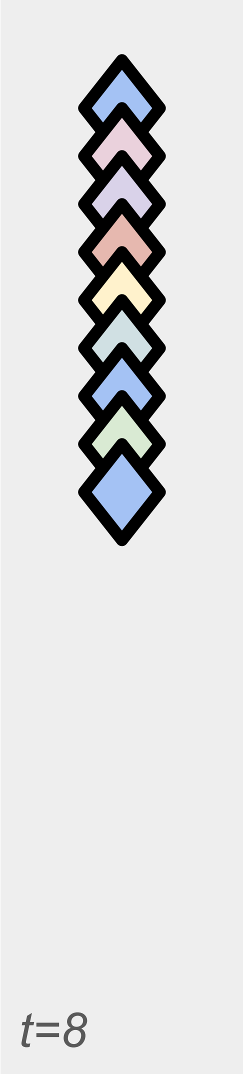


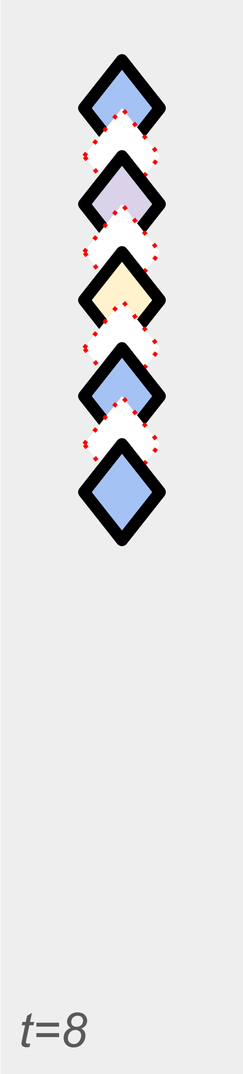
end-state

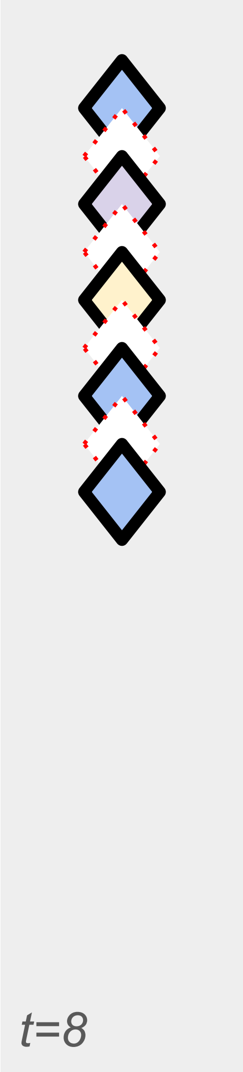


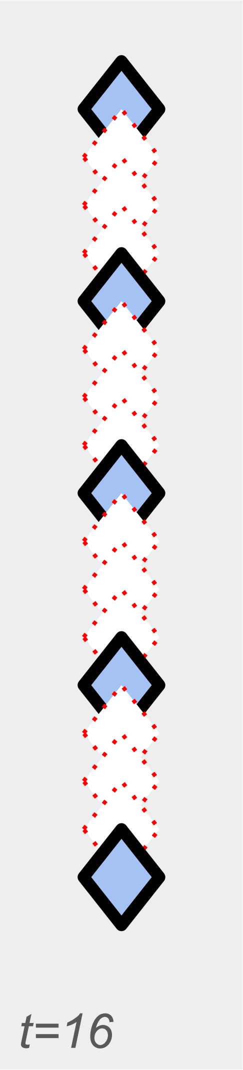
reconstruct

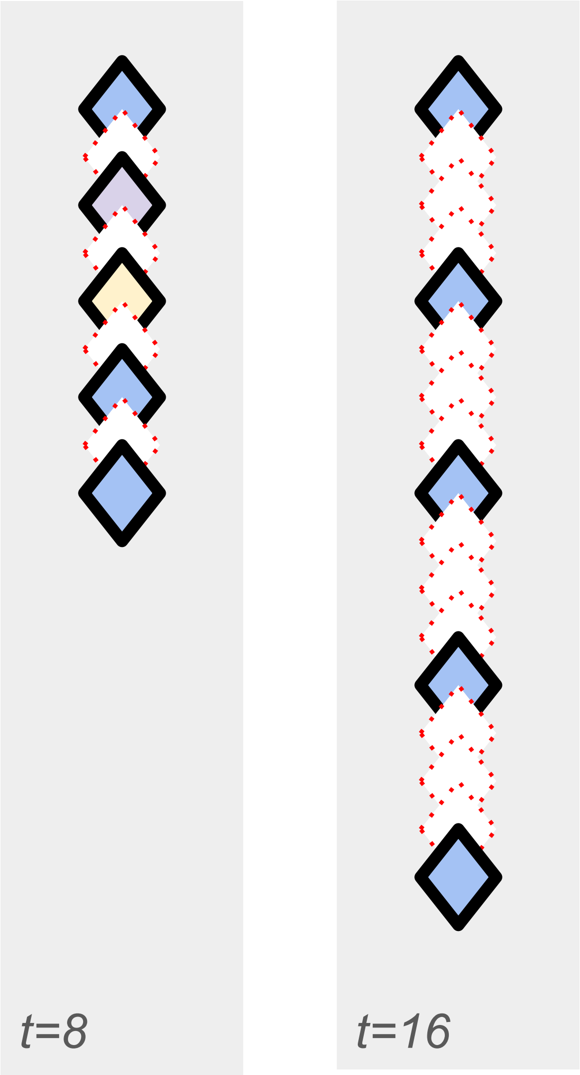


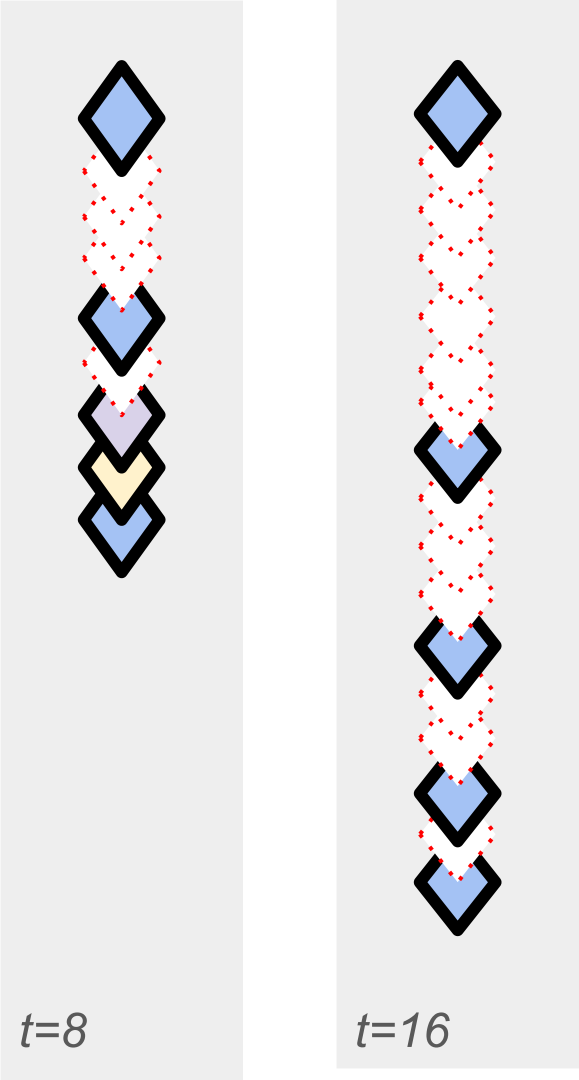

t=8

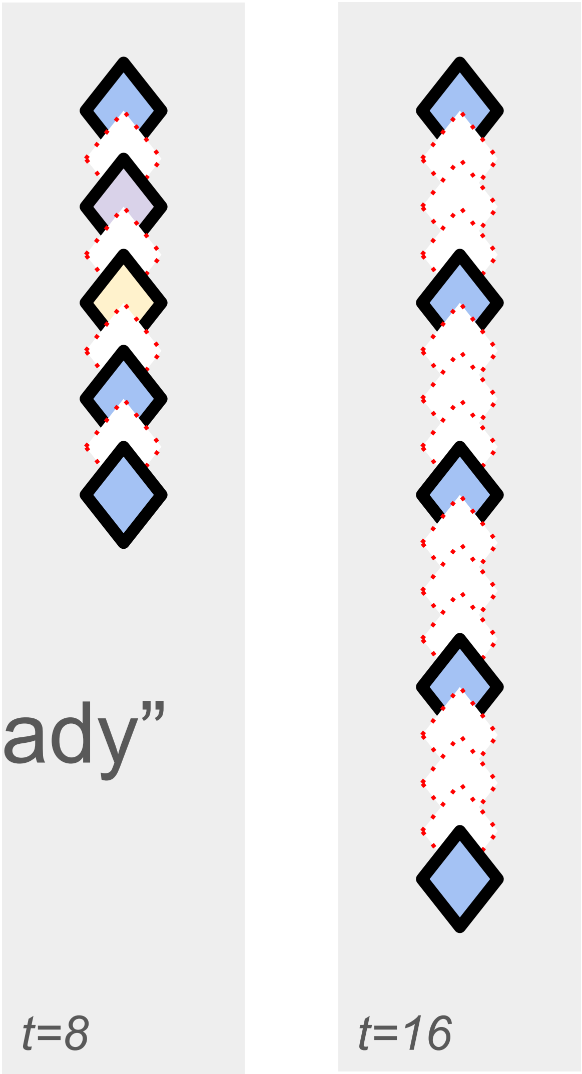

t=8


t=8

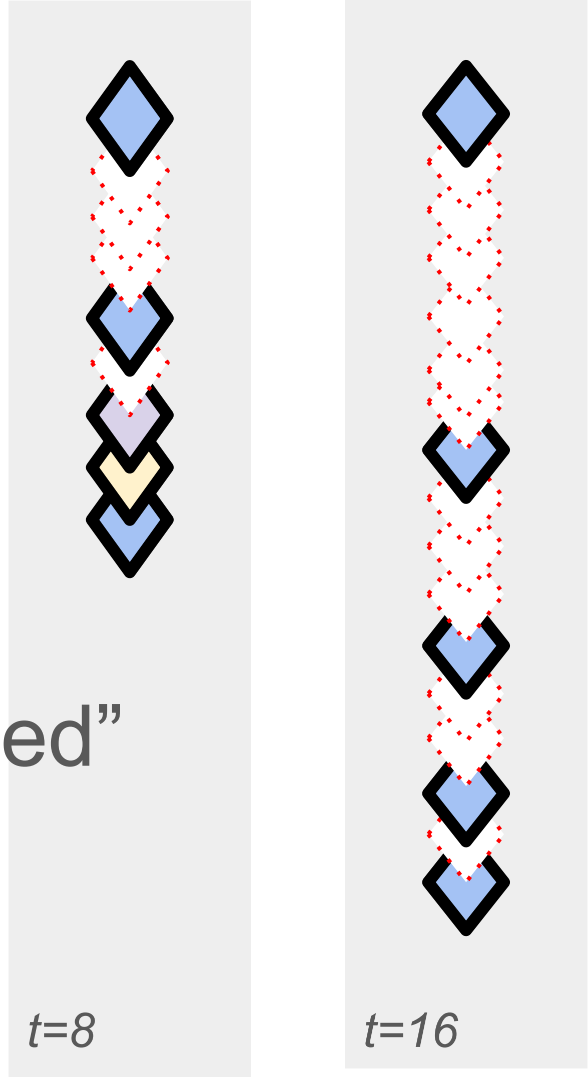

t=16


t=8
t=16

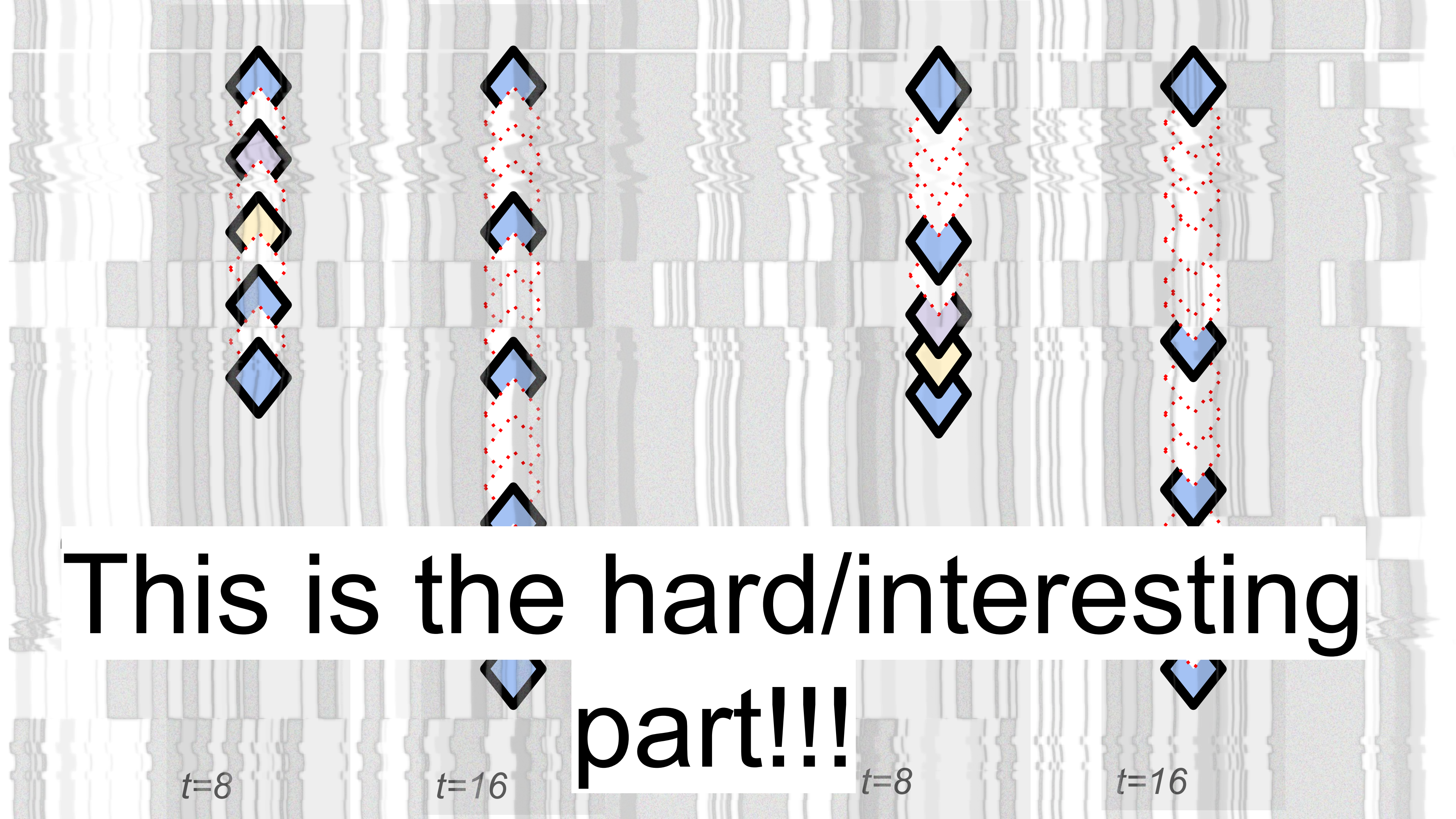

t=8
t=16



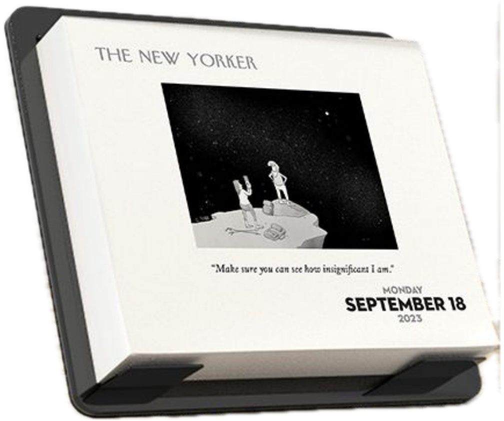
“steady”



“tilted”



This is the hard/interesting part!!!



data stream

Generalized Ring Buffer (Gunther, 2014)

data storage

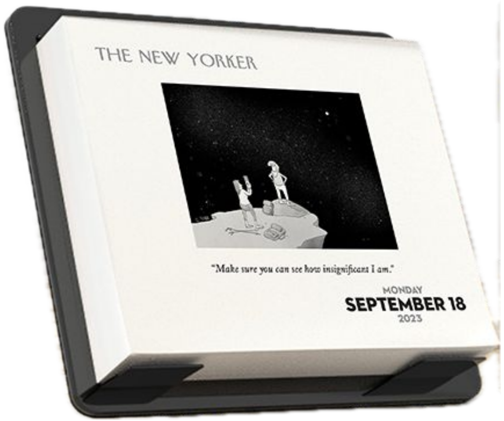
data stream

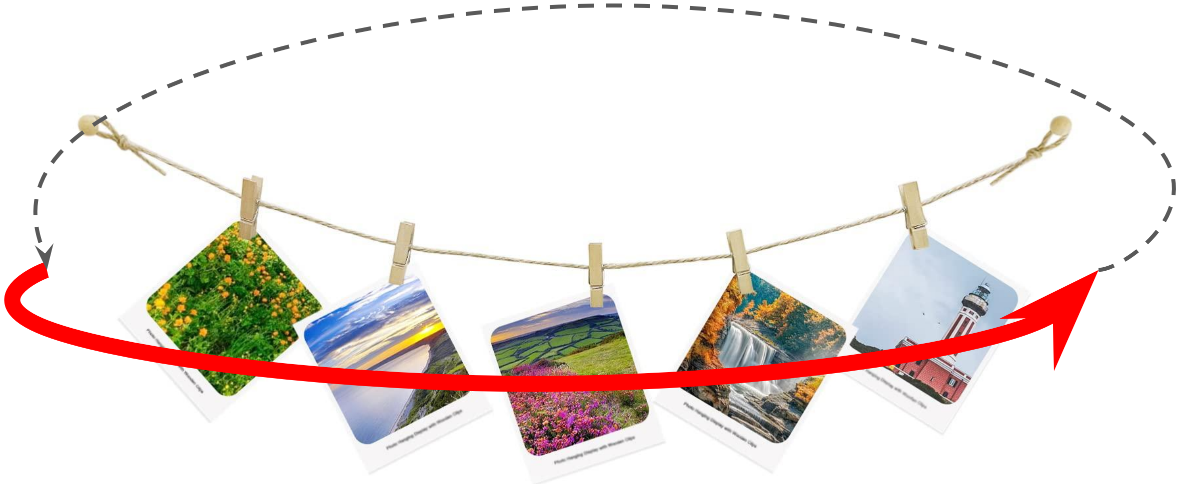
It is generation 72…

… put at position 4

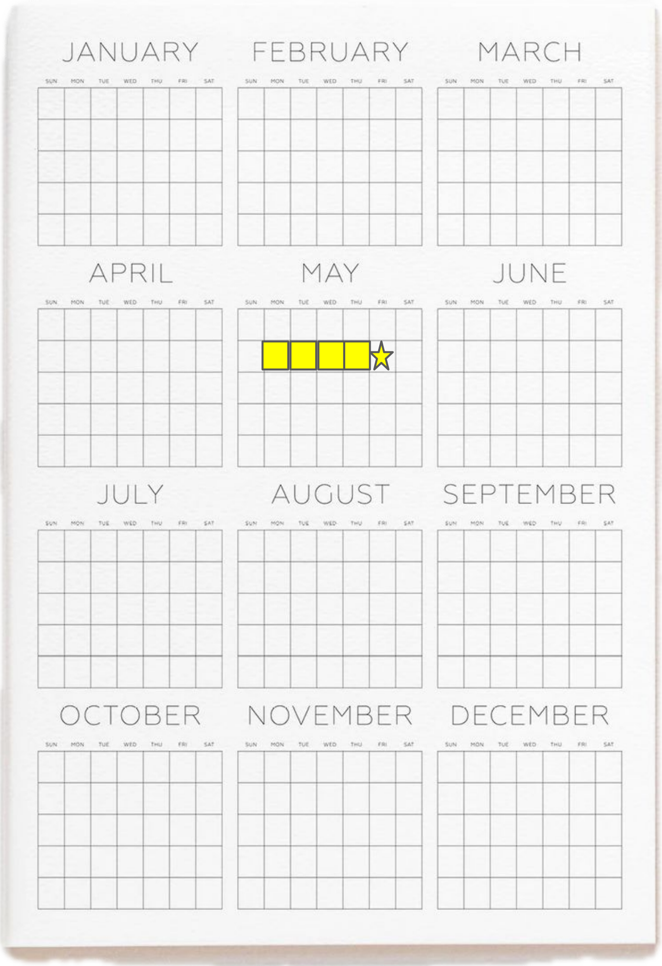
data storage

Simple Ring Buffer



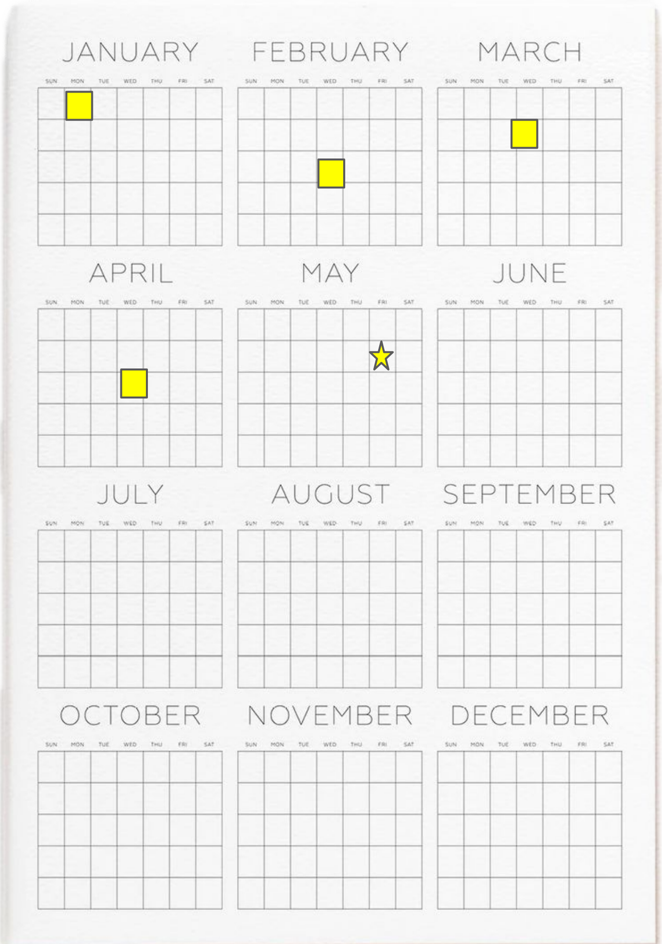


temporal coverage



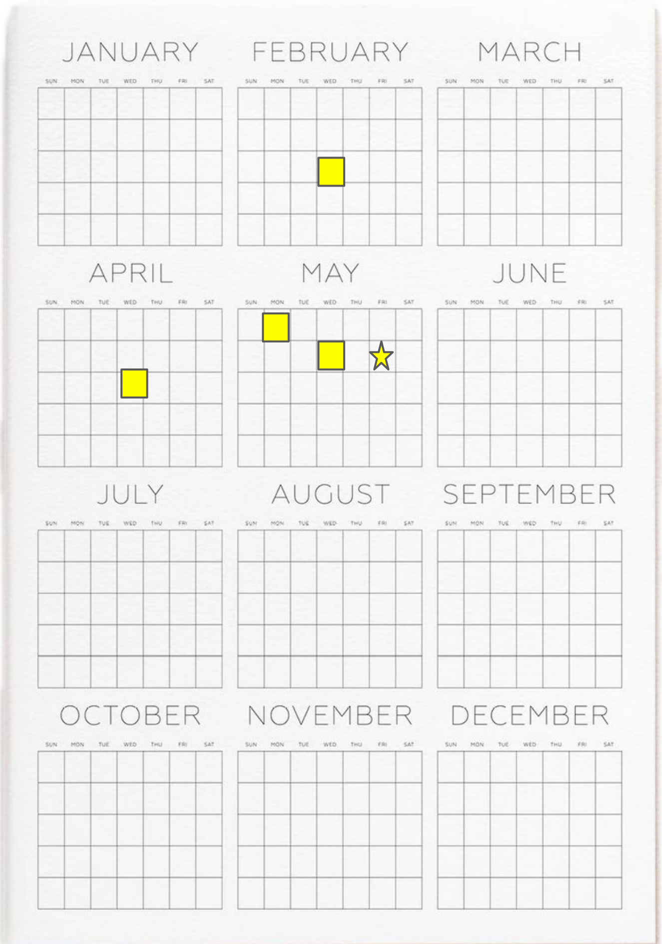
Steady
Retention

temporal coverage

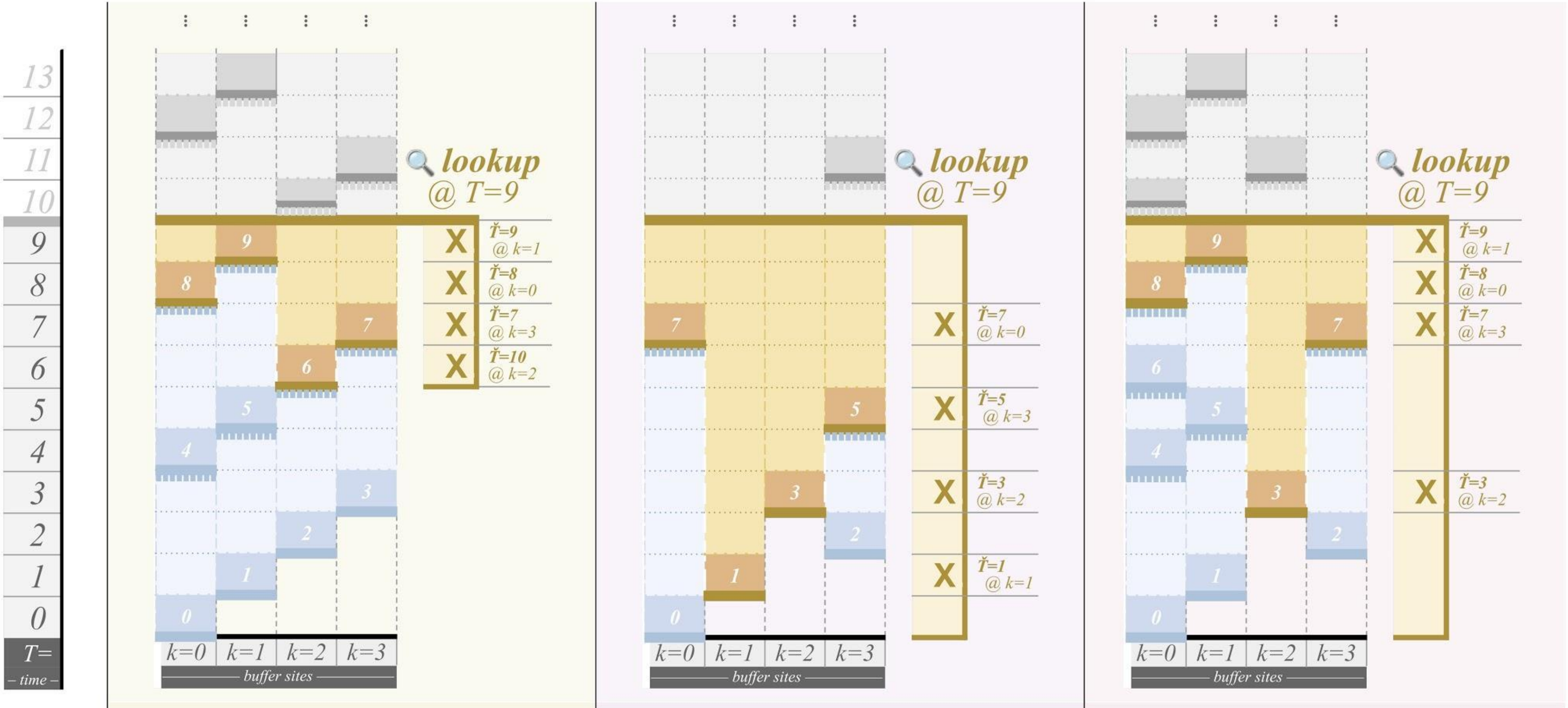


Tilted
Retention

temporal coverage

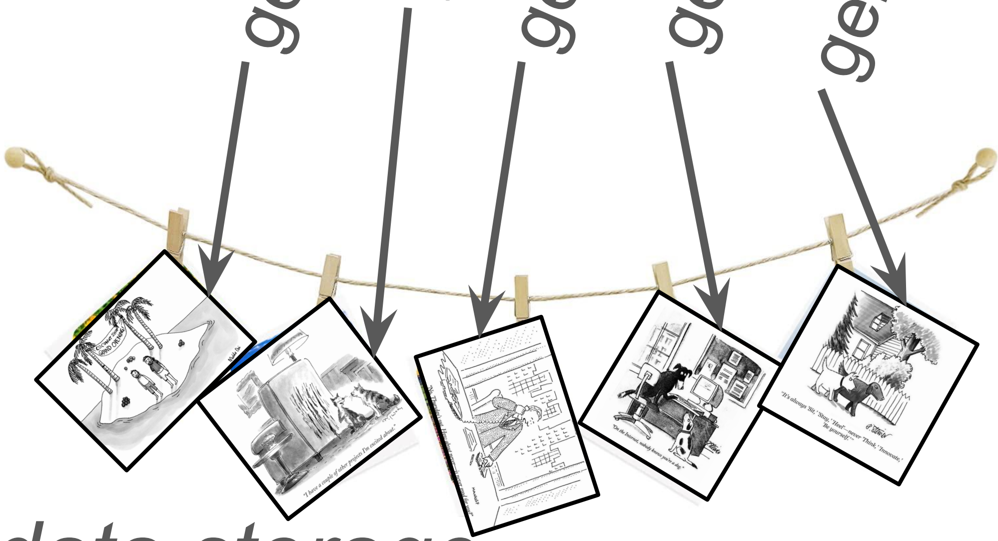


Generalized Ring Buffer (Gunther, 2014)



(a) simple ring buffer **(b) generalized ring buffer: steady curation** **(c) generalized ring buffer: tilted curation**

72 generations have elapsed…



gen 68

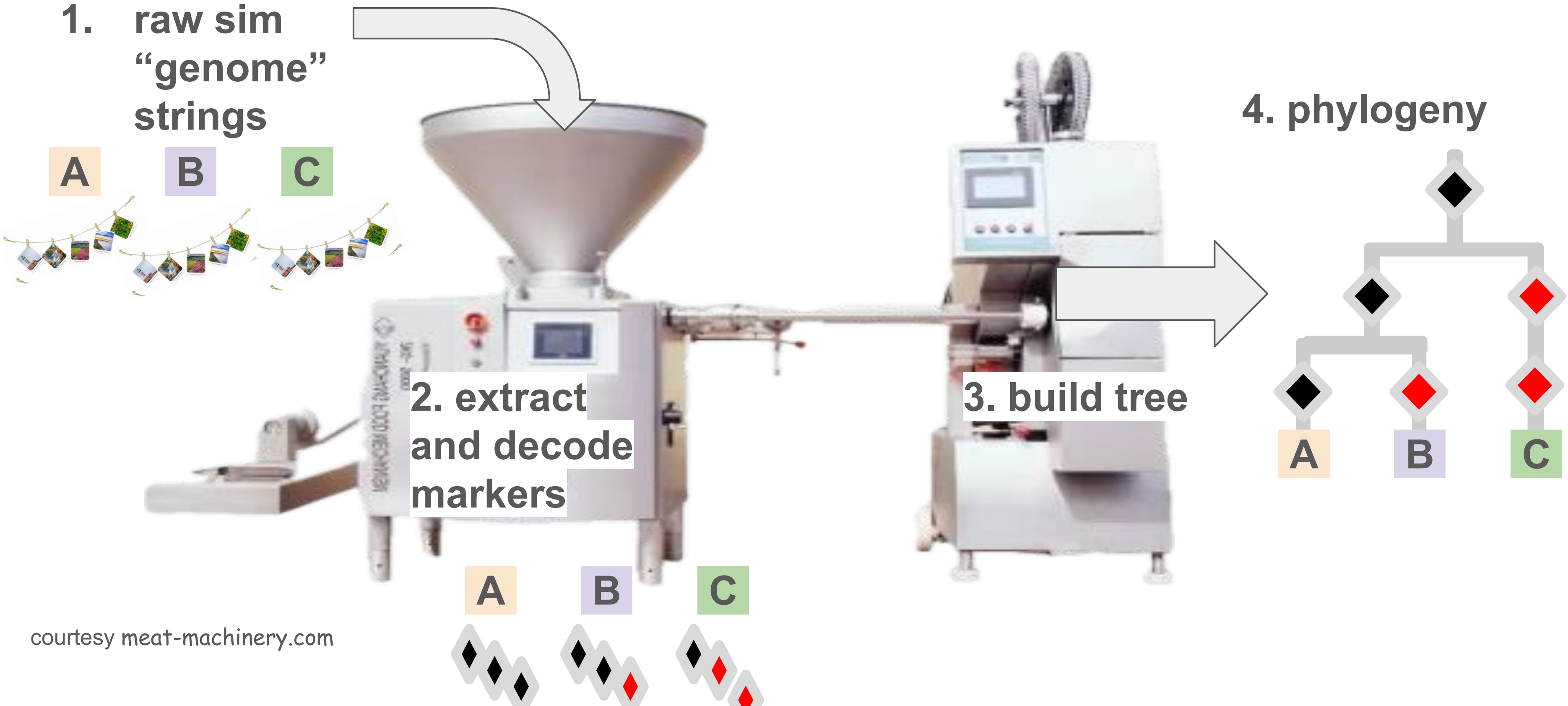
gen 54

gen 45

gen 23

gen 61

data storage



1. raw sim “genome” strings

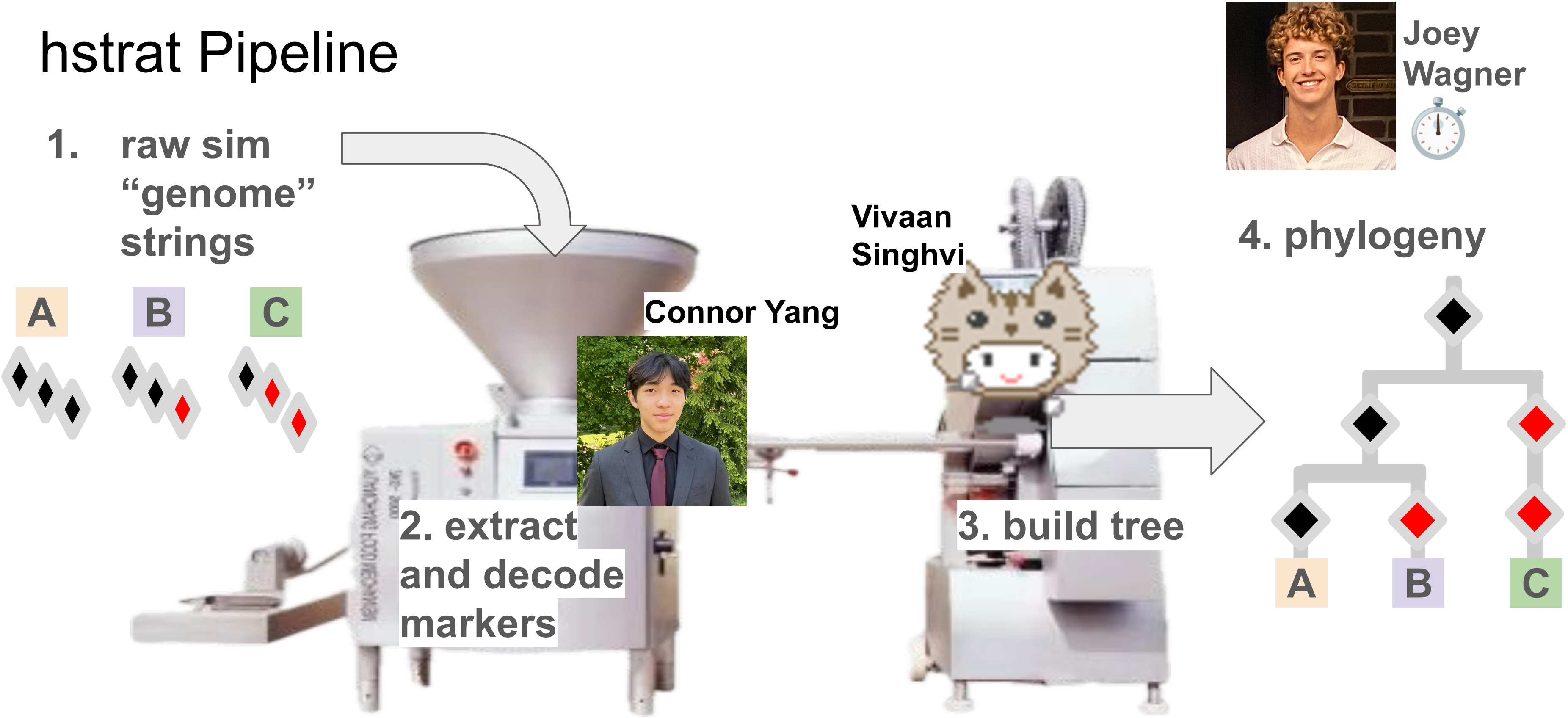
2. extract and decode markers

3. build tree

4. phylogeny

courtesy meat-machinery.com

hstrat Pipeline



1. raw sim "genome" strings

A B C

Connor Yang

2. extract and decode markers

Vivaan Singhvi

3. build tree

Joey Wagner ⏱️

4. phylogeny

A B C

courtesy meat-machinery.com

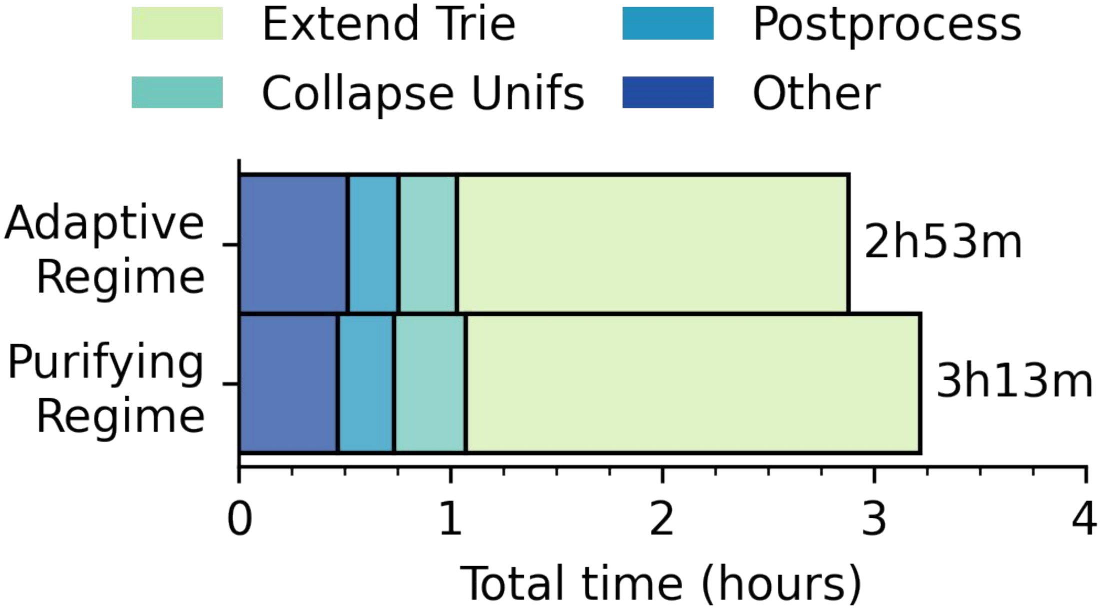
hstrat Pipeline

1 billion tip phylogeny

courtesy meat-machinery.com

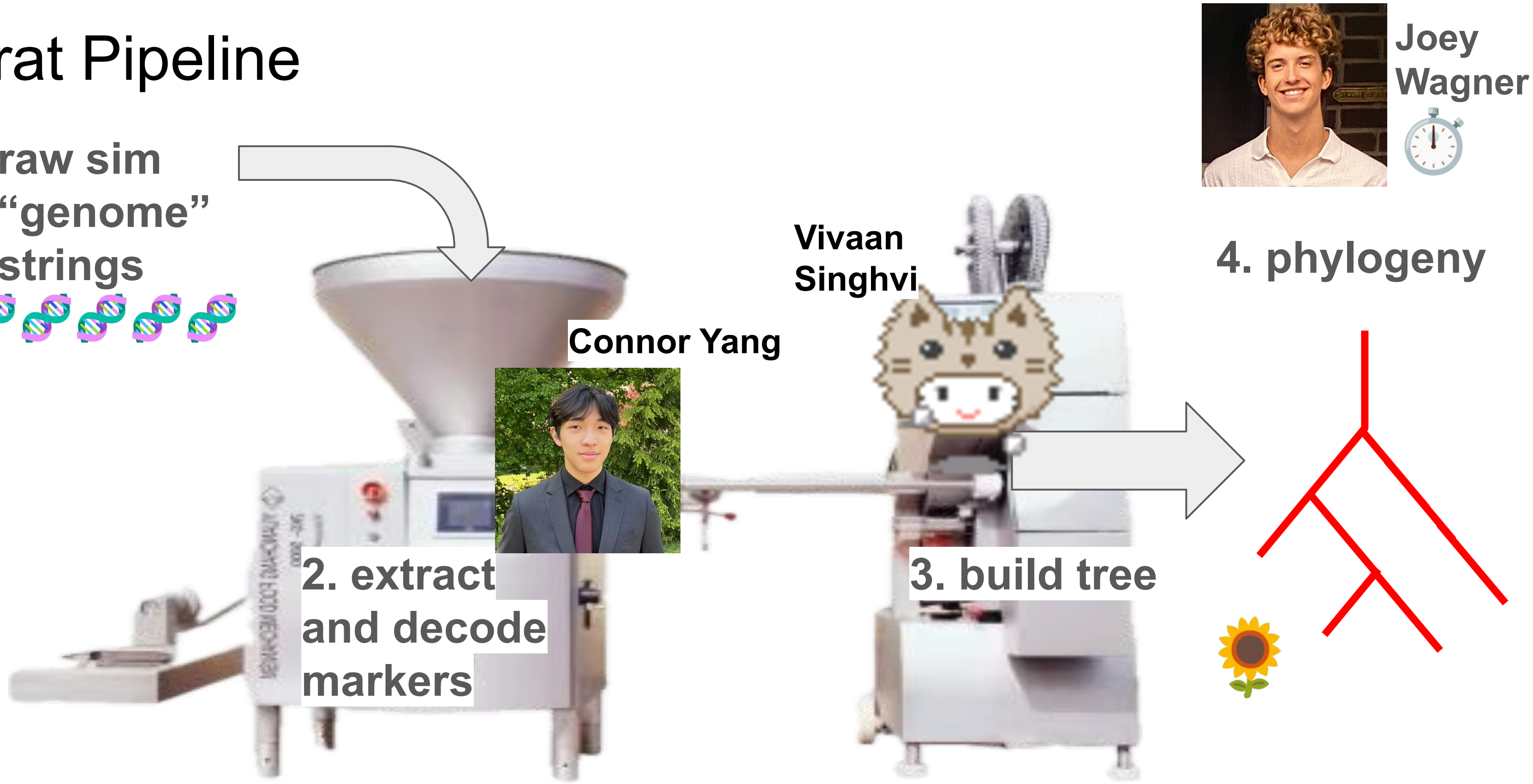
Joey Wagner

⏱️



hstrat Pipeline

1. raw sim “genome” strings 🧬🧬🧬🧬🧬🧬🧬



2. extract and decode markers

Connor Yang

3. build tree

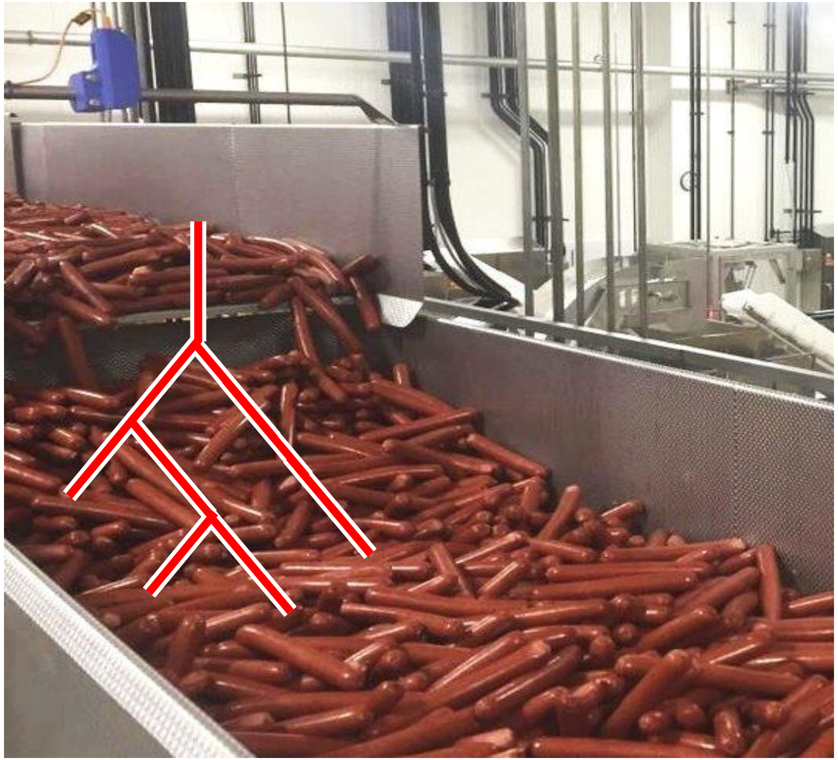
Vivaan Singhvi

4. phylogeny 🌻

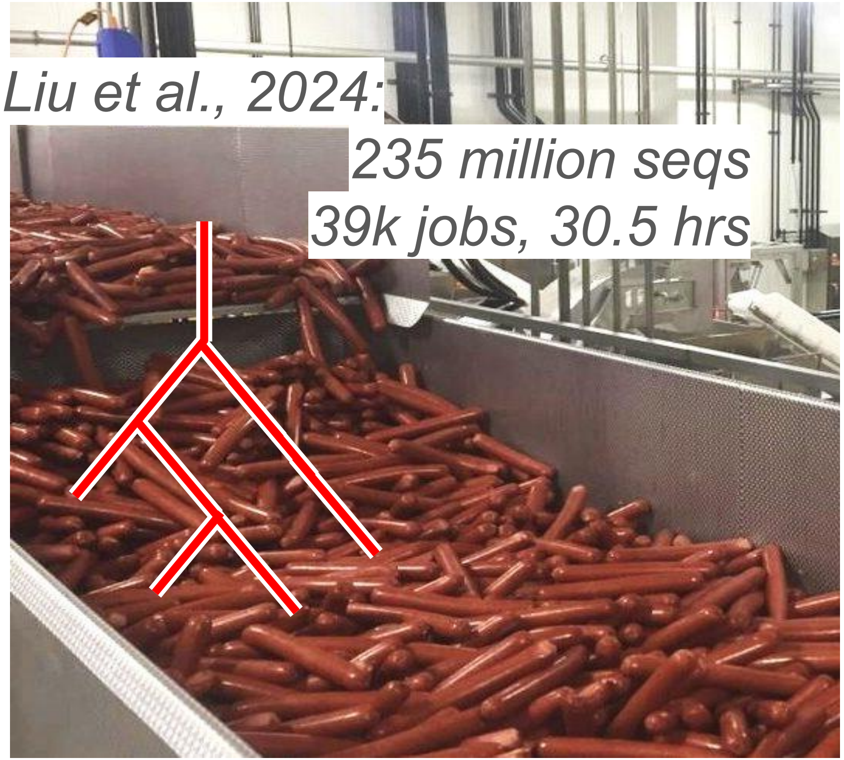
Joey Wagner ⏱️

courtesy meat-machinery.com

high-throughput phylogeny generation (*in vivo* & *in silico*)



high-throughput phylogeny generation (*in vivo* & *in silico*)

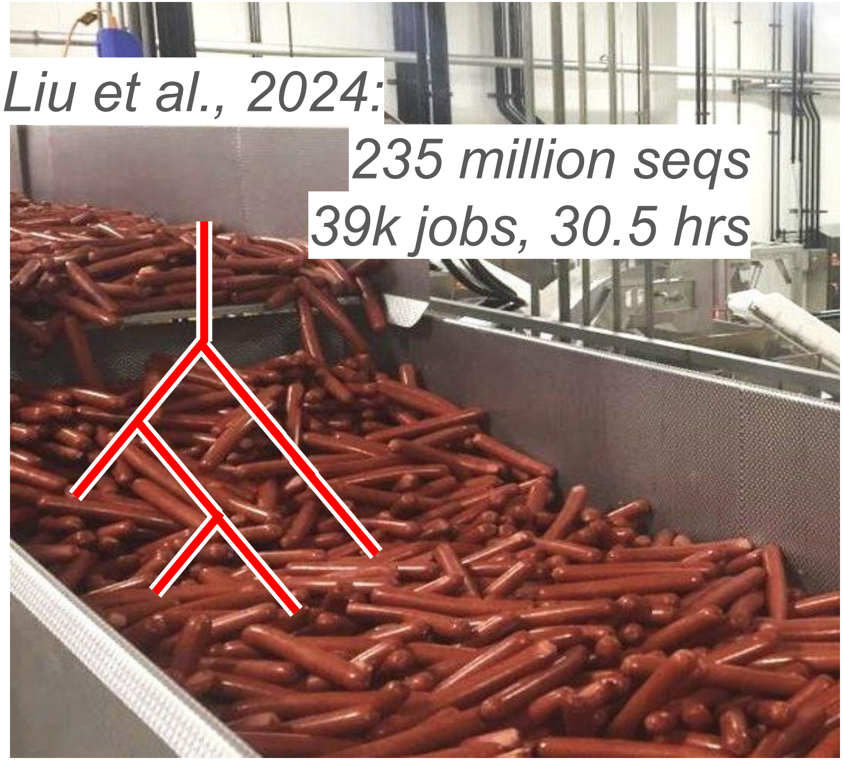


Liu et al., 2024:

235 million seqs

39k jobs, 30.5 hrs

high-throughput phylogeny generation (*in vivo* & *in silico*)



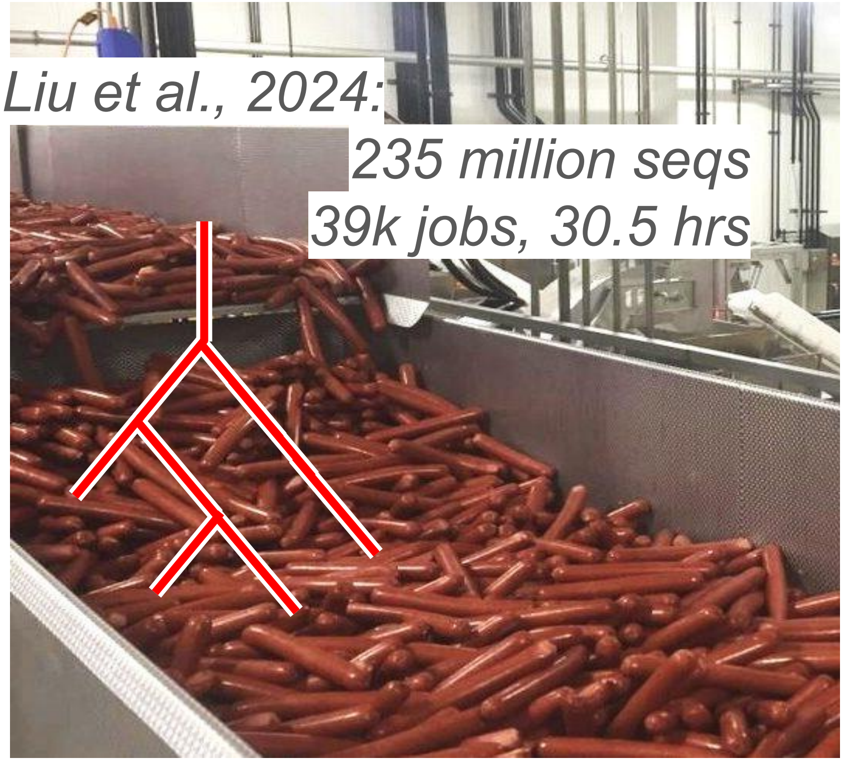
Liu et al., 2024:

235 million seqs

39k jobs, 30.5 hrs

existing phylogeny analyses and tools

high-throughput phylogeny generation (*in vivo* & *in silico*)

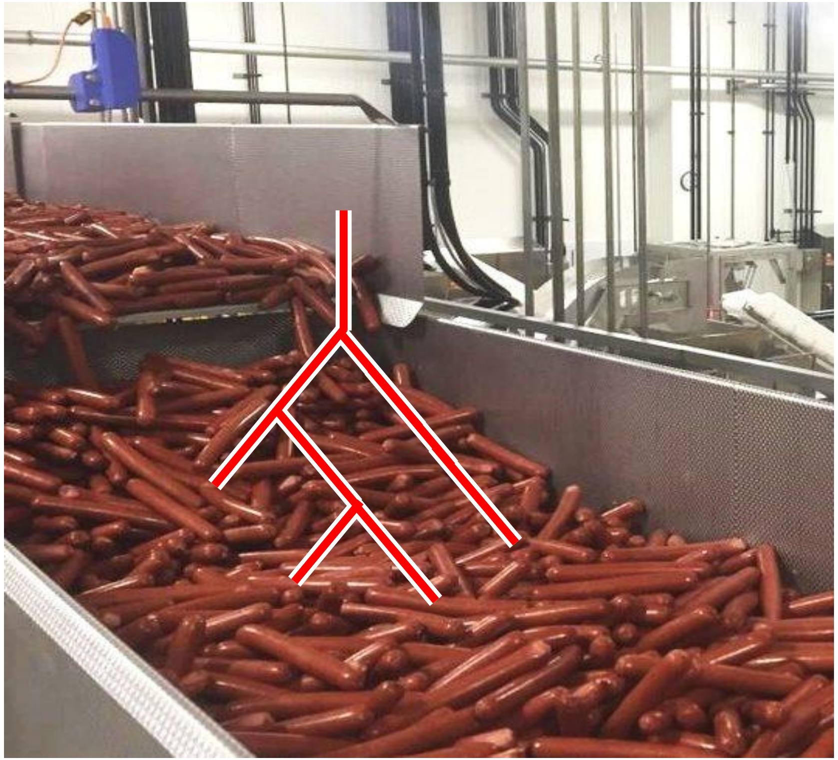


Liu et al., 2024:

235 million seqs

39k jobs, 30.5 hrs

high-throughput phylogeny generation (*in vivo* & *in silico*)



contemporary analyses

use more phylogenies

use larger phylogenies

DataFrame tools for big phylogeny data

https://github.com/mmore500/phyloframe

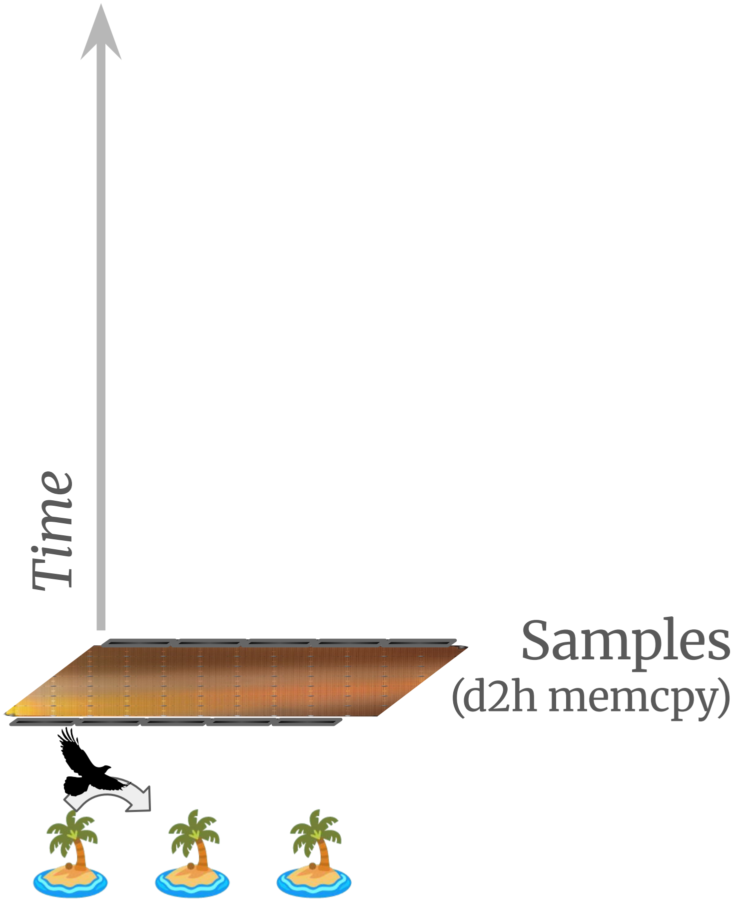
Alife data standard*

```
T0 = (((A:0.12,B:0.14):0.32,C:0.81):0.34,(D:0.91,(E:0.63,F:0.40):0.23):0.21);
T1 = (((A:0.32,B:0.12):0.42,D:0.04):0.31,(C:0.62,(E:0.82,F:0.92):0.33):0.15);
T2 = (((A:0.24,B:0.63):0.31,E:0.72):0.43,(D:0.01,(C:0.22,F:0.33):0.11):0.46);
```

Part II:

Cerebras SDK Workflow on Wafer-Scale Cloud

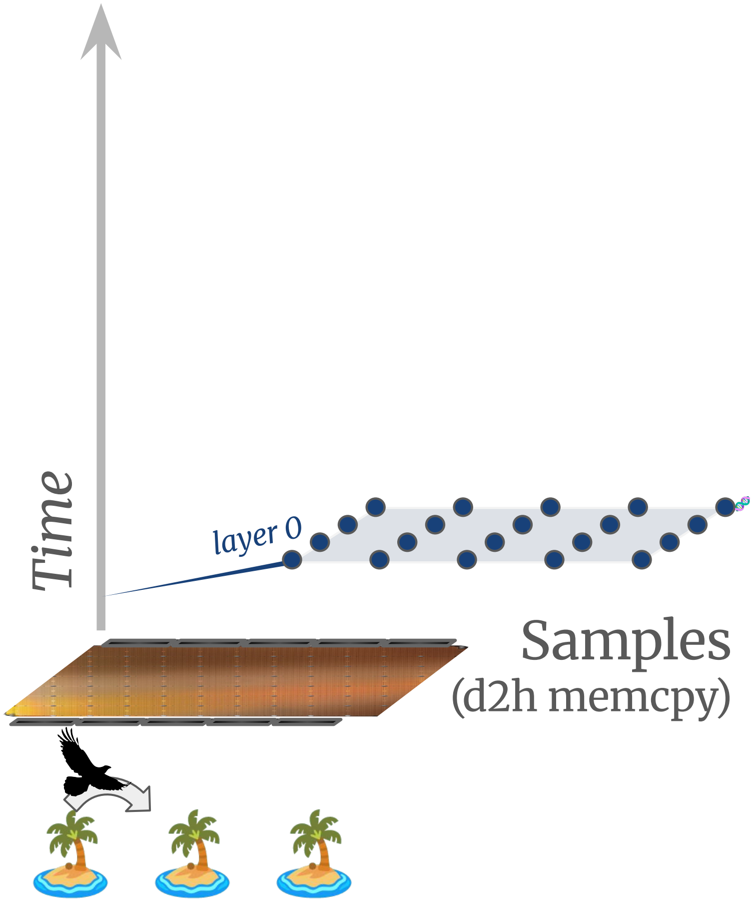
morenoma@umich.edu



Time

Samples

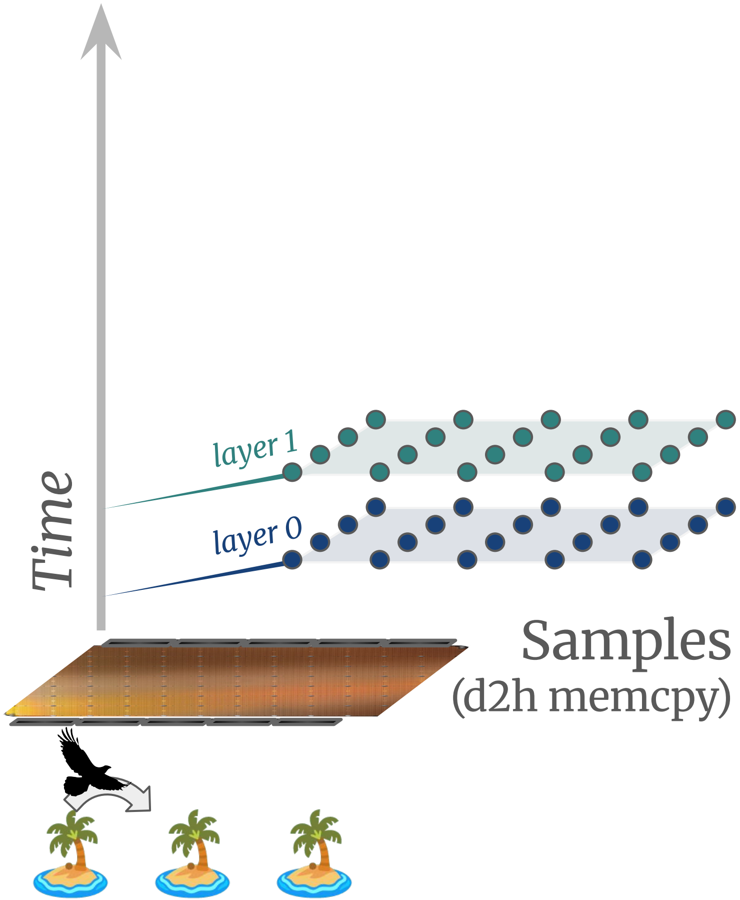
(d2h memcpy)



Time

layer 0

Samples
(d2h memcpy)



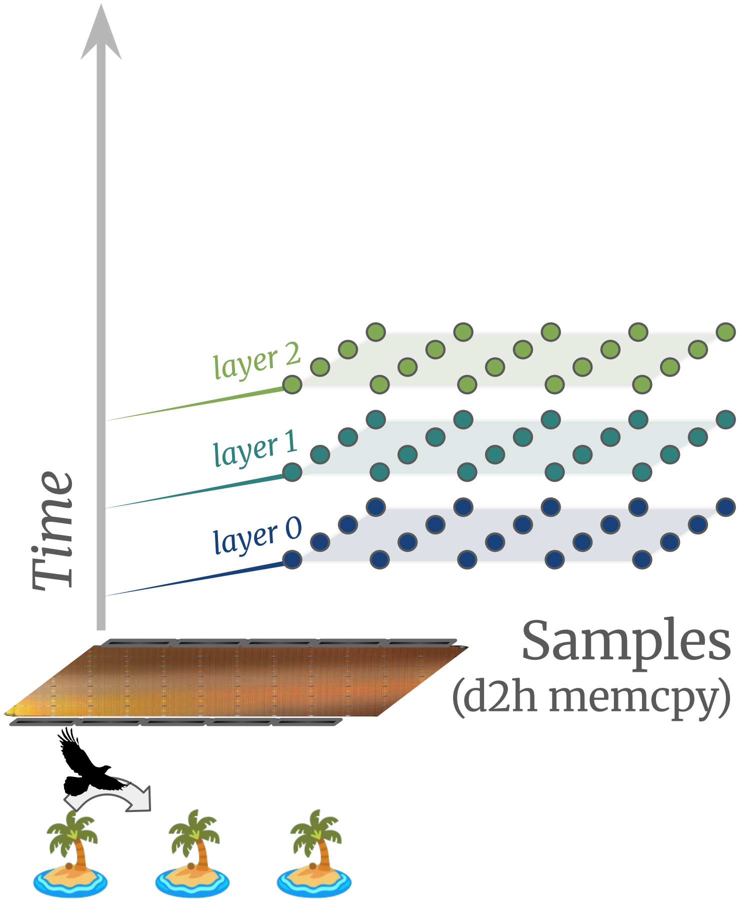
Time

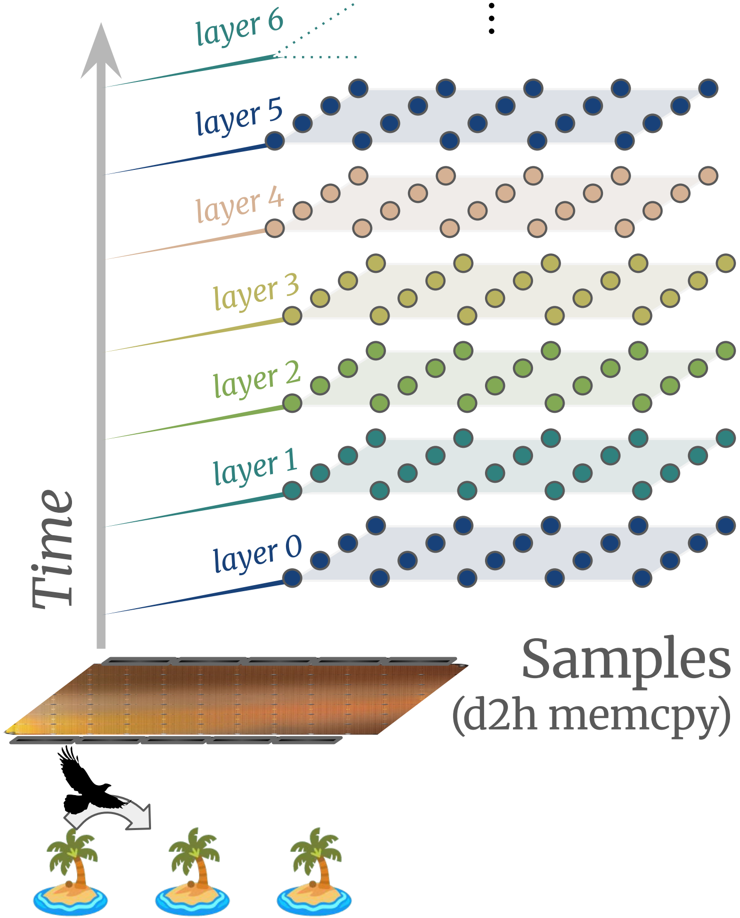
layer 1

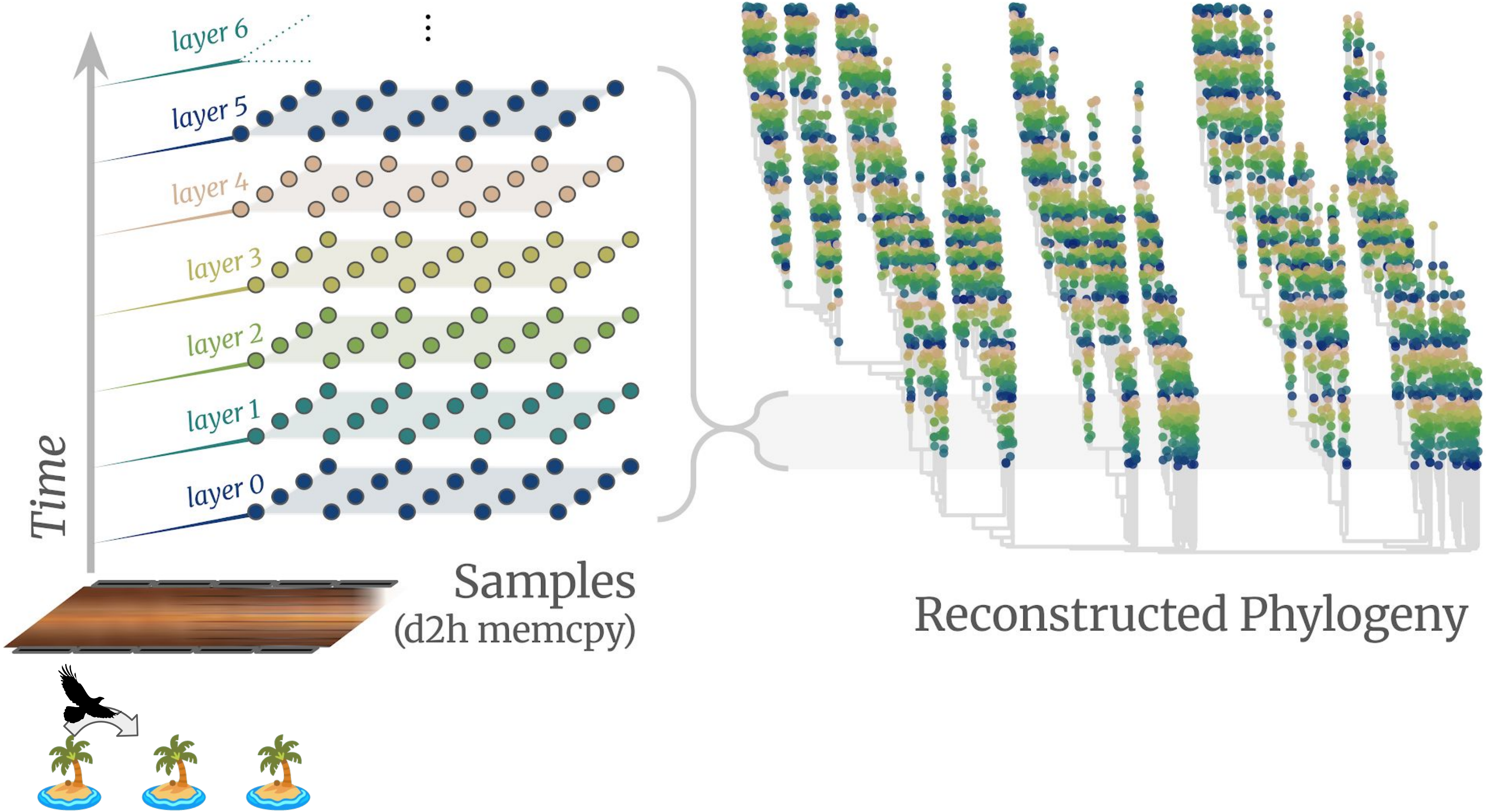
layer 0

Samples

(d2h memcpy)


Time
layer 2
layer 1
layer 0
Samples
(d2h memcpy)


layer 6
⋮
layer 5
layer 4
layer 3
layer 2
layer 1
layer 0
Time
Samples
(d2h memcpy)


layer 6
layer 5
layer 4
layer 3
layer 2
layer 1
layer 0
Time
Samples
(d2h memcpy)
Reconstructed Phylogeny

login node

compile &
submit job

https://github.com/mmore500/wse-async-ga

CS-3

~8,750 generations / sec
1,250 memcpy layers kept from ~9,250 cycles

on-chip (~10 minutes)

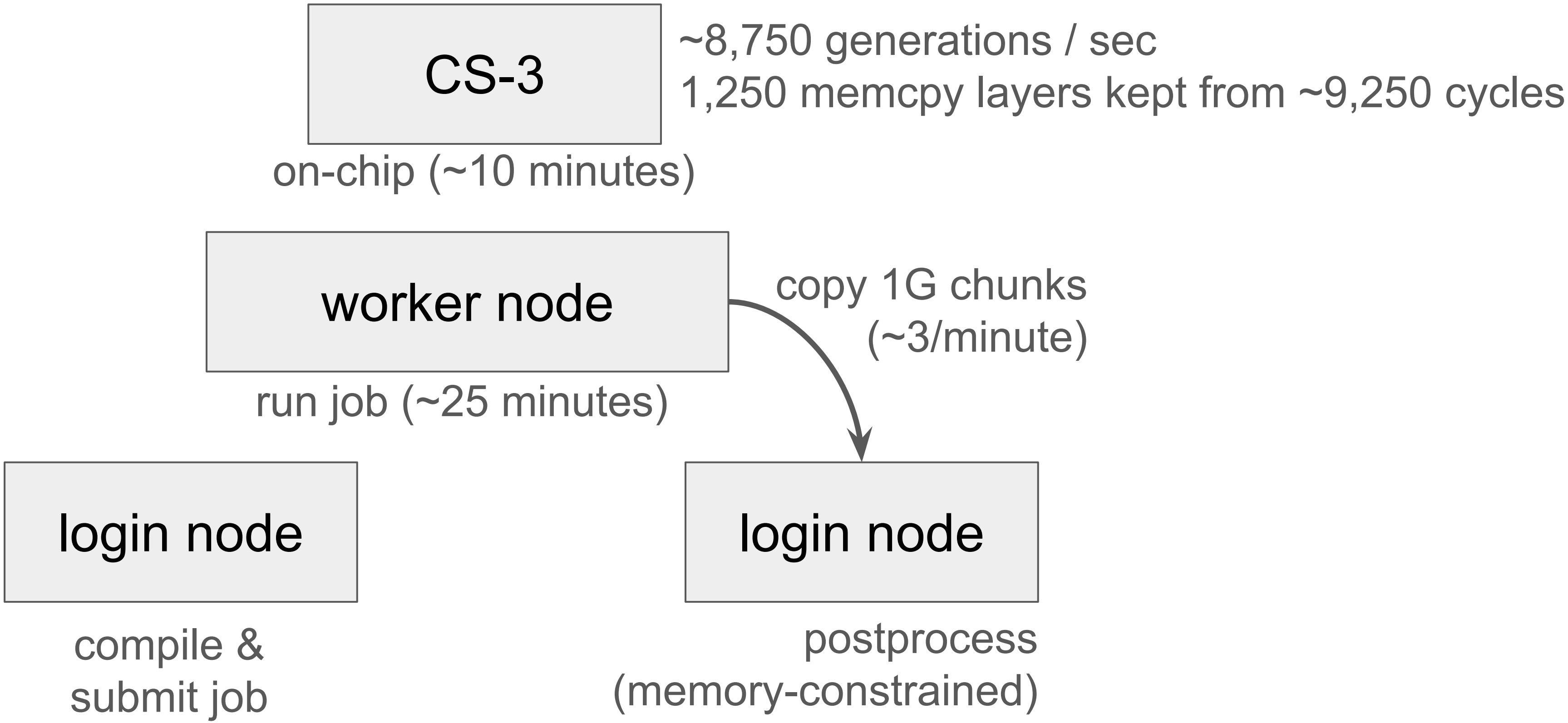
worker node

run job (~25 minutes)

login node

compile &
submit job

https://github.com/mmore500/wse-async-ga



CS-3

~8,750 generations / sec

1,250 memcpy layers kept from ~9,250 cycles

on-chip (~10 minutes)

worker node

run job (~25 minutes)

copy 1G chunks
(~3/minute)

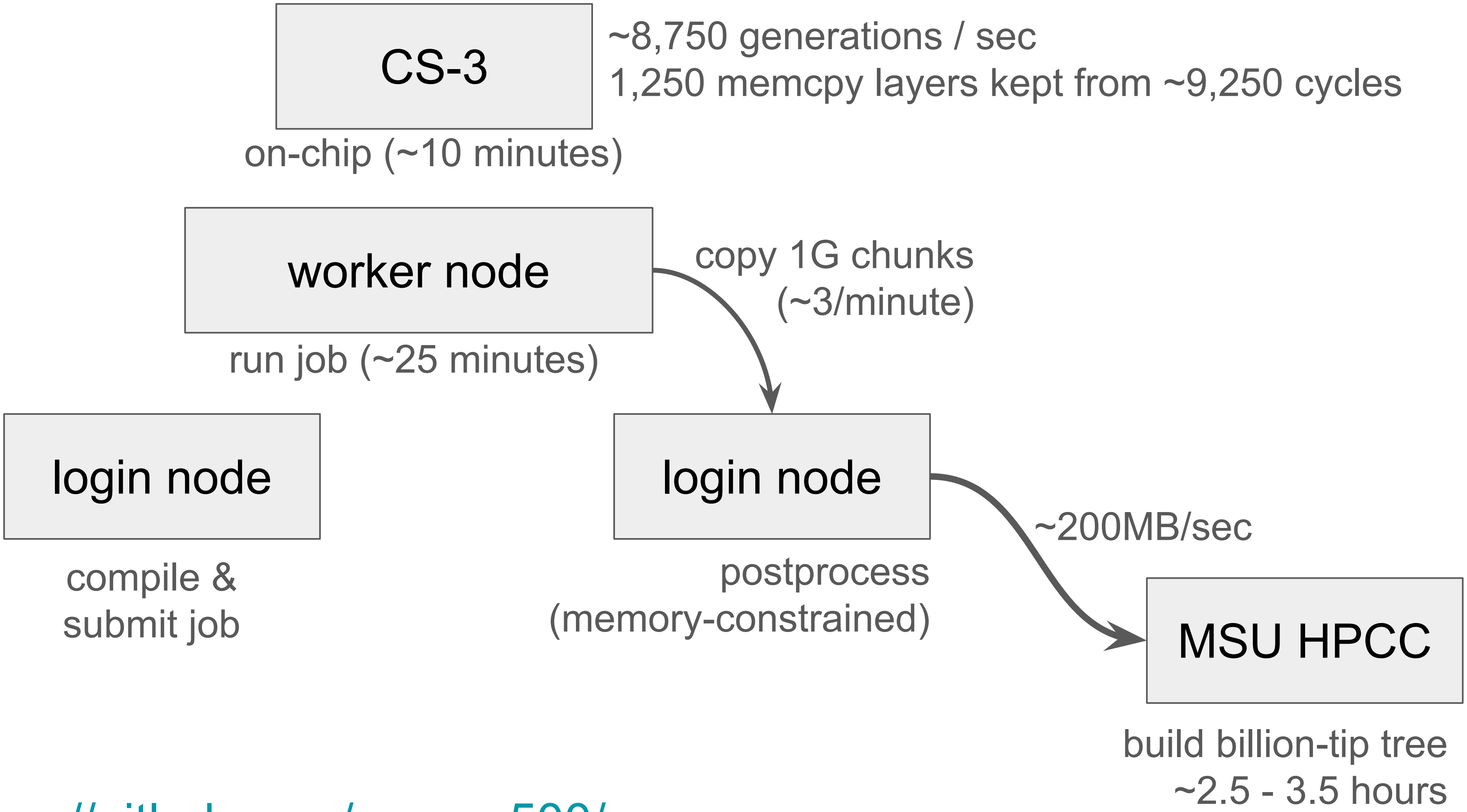
login node

compile &
submit job

login node

postprocess
(memory-constrained)

https://github.com/mmore500/wse-async-ga



CS-3

~8,750 generations / sec
1,250 memcpy layers kept from ~9,250 cycles

on-chip (~10 minutes)

worker node

run job (~25 minutes)

copy 1G chunks
(~3/minute)

login node

compile &
submit job

login node

postprocess
(memory-constrained)

~200MB/sec

MSU HPCC

build billion-tip tree
~2.5 - 3.5 hours

https://github.com/mmore500/wse-async-ga

CS-3

~8,750 generations / sec
1,250 memcpy layers kept from ~9,250 cycles

on-chip (~10 minutes)

worker node

run job (~25 minutes)

copy 1G chunks
(~3/minute)

login node

compile &
submit job

login node

postprocess
(memory-constrained)

~200MB/sec

MSU HPCC

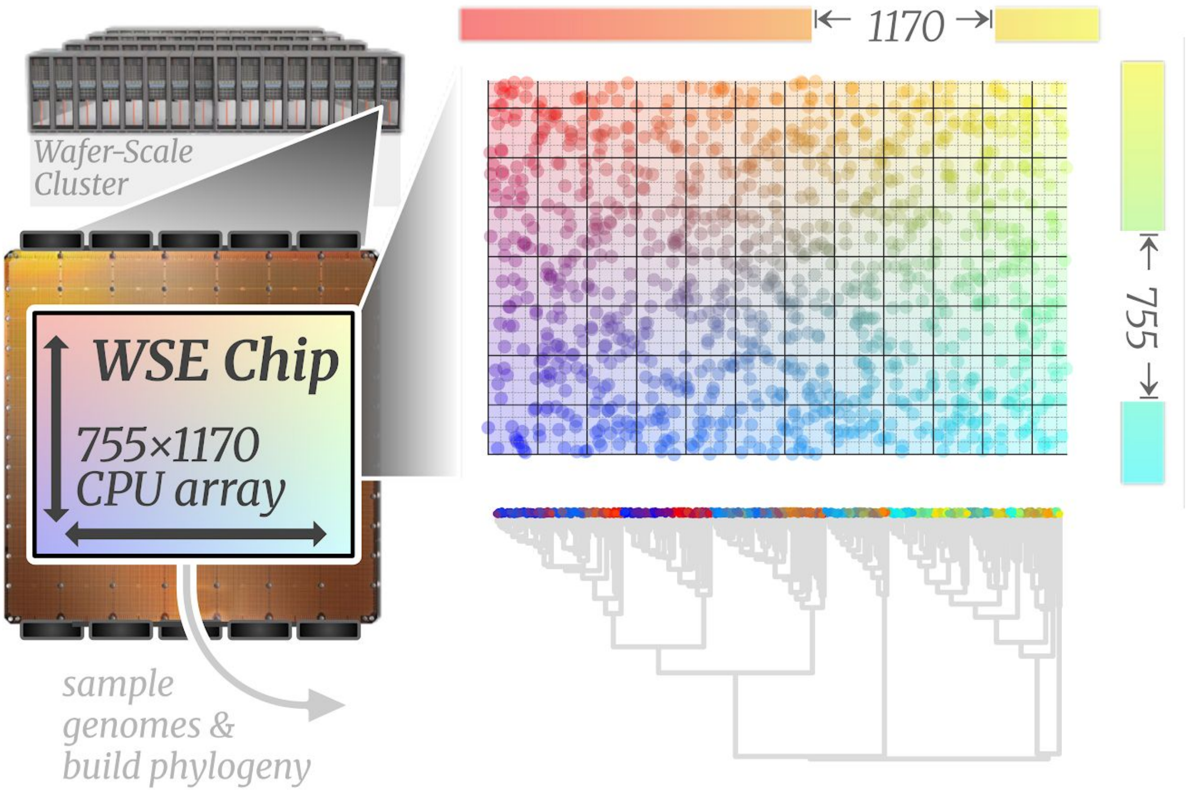
build billion-tip tree
~2.5 - 3.5 hours

https://github.com/mmore500/wse-async-ga

Workflow Bottlenecks/Challenges

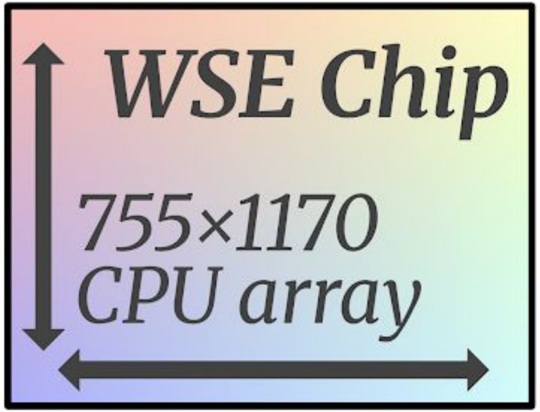
- limited memory available for postprocessing on login node
- constrained bandwidth for data egress from WSE job to login node
  - WSE chip can’t be relinquished while copying data
- moderate bandwidth for data egress from login node to home HPC
- SDK v1.4.0 container still runs Python3.8
  - would love newer Python with Polars

Workflow Preliminary Results



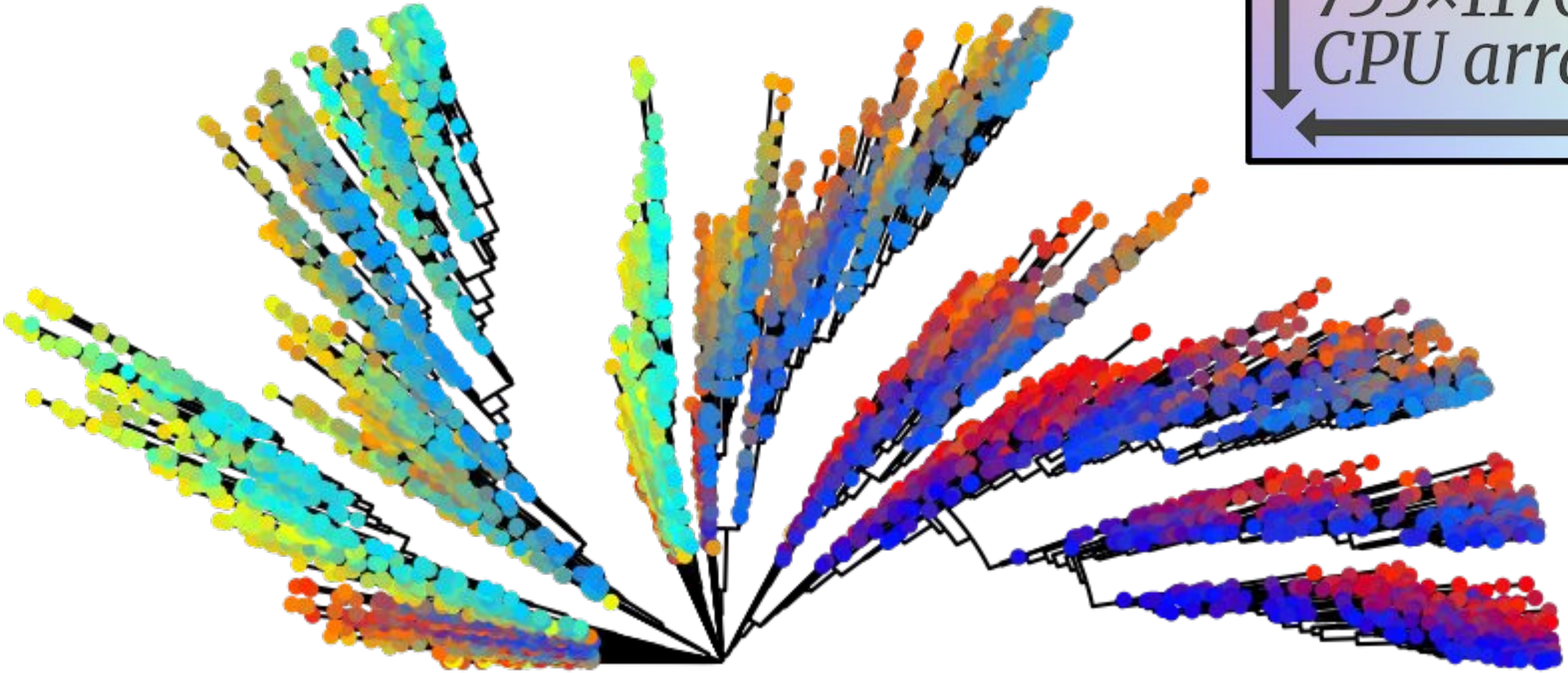
(subsampled from 1B tip reconstruction)

Workflow Preliminary Results



WSE Chip

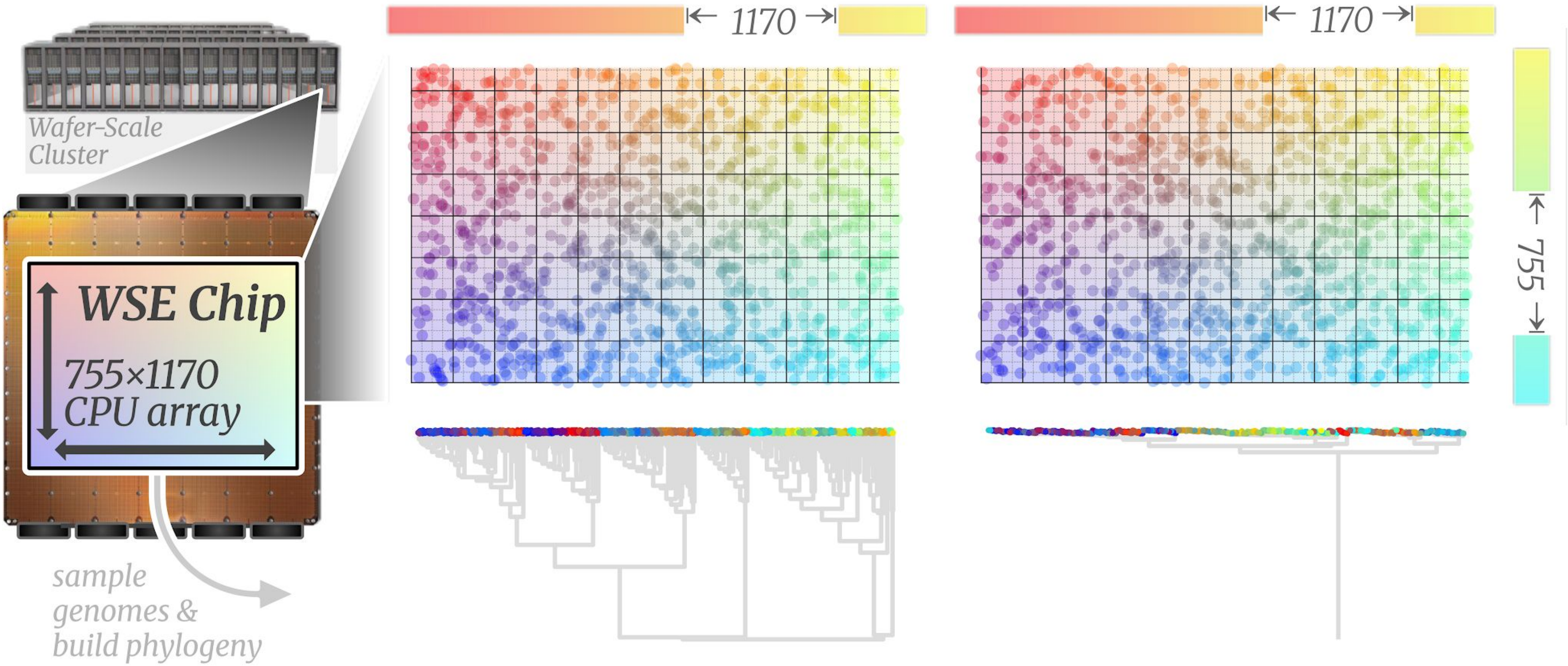
755×1170
CPU array



Workflow Preliminary Results

(subsampled from 1B tip reconstruction)

Workflow Preliminary Results

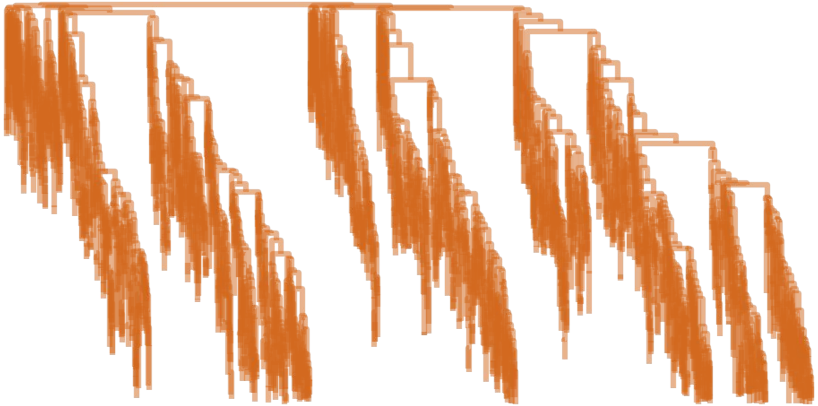


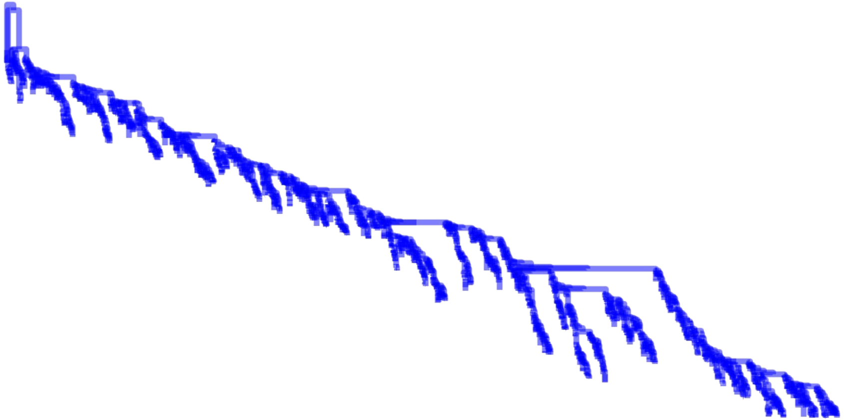
(subsampled from 1B tip reconstruction)

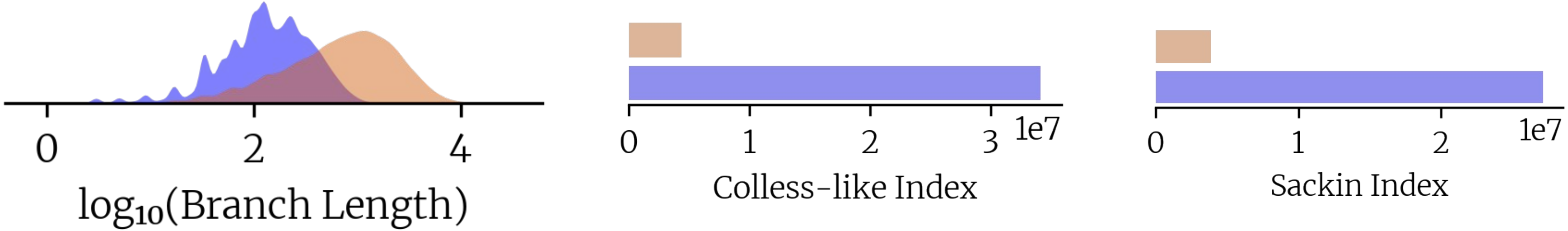
Workflow Preliminary Results

Neutral Conditions

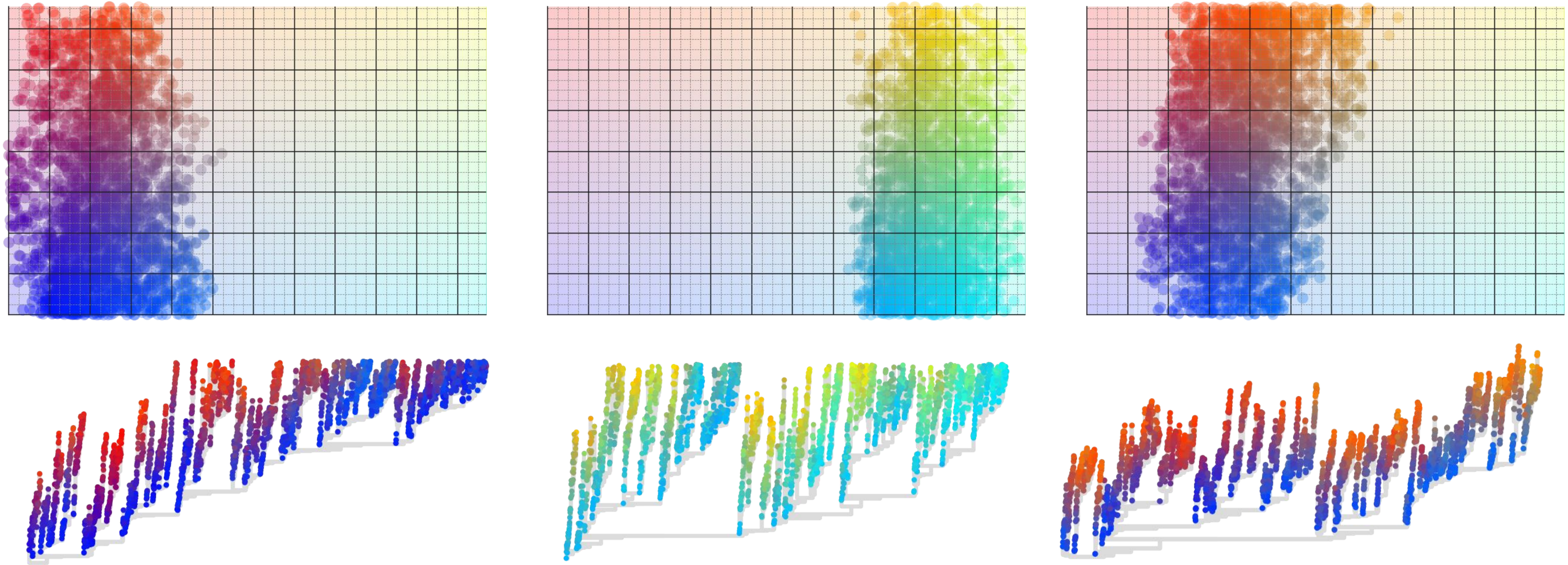
Adaptive Conditions



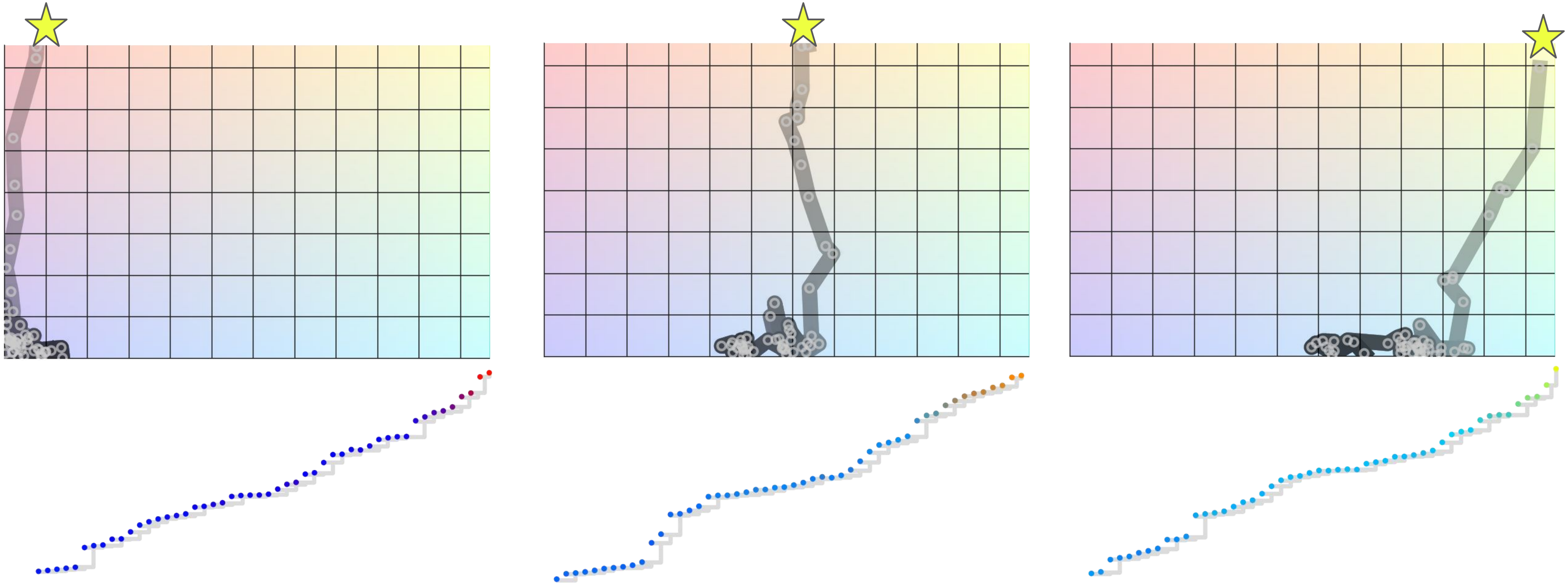




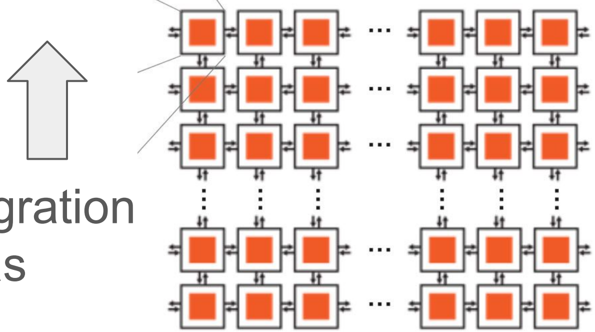
Workflow Preliminary Results



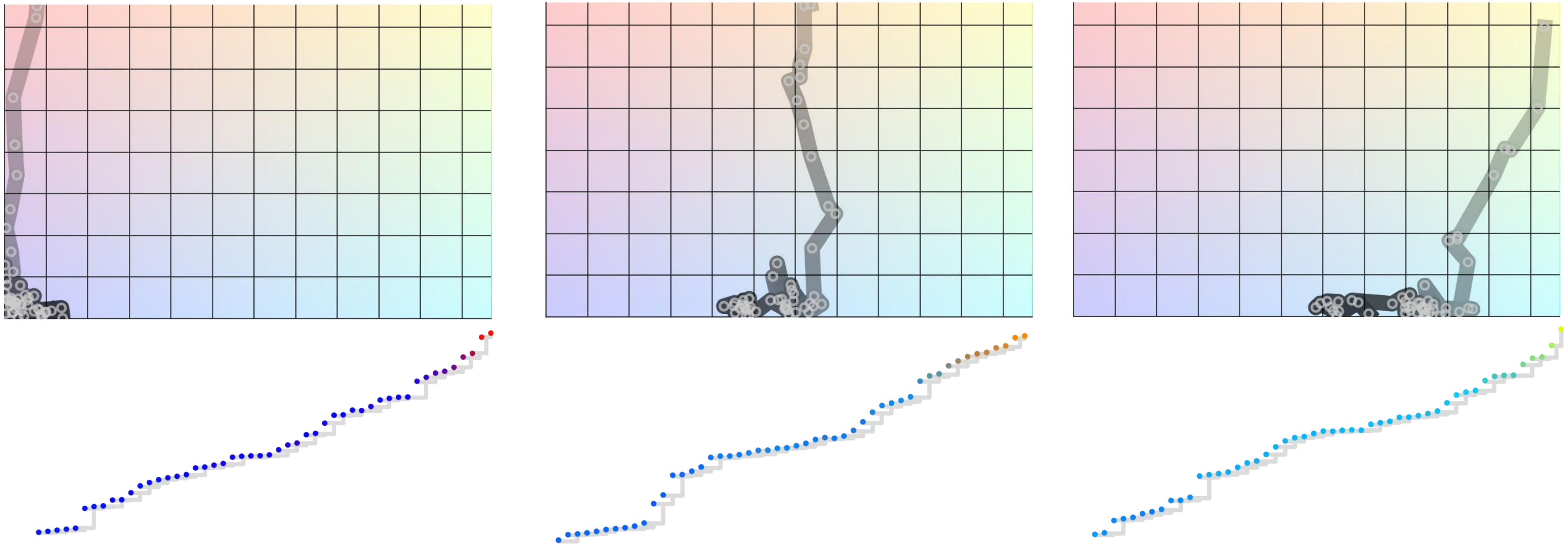
Workflow Preliminary Results

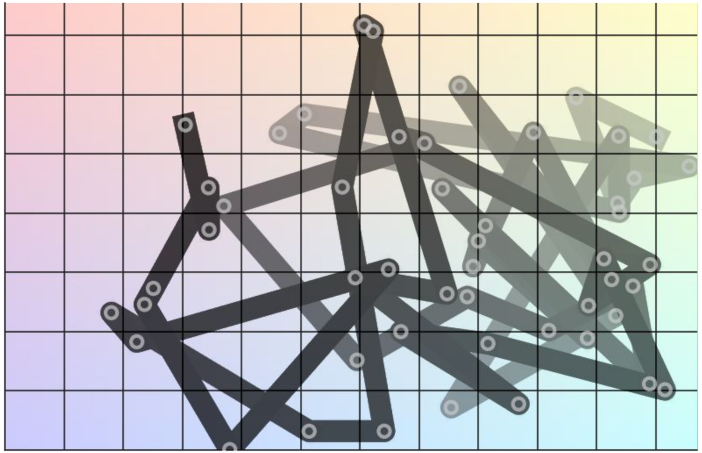


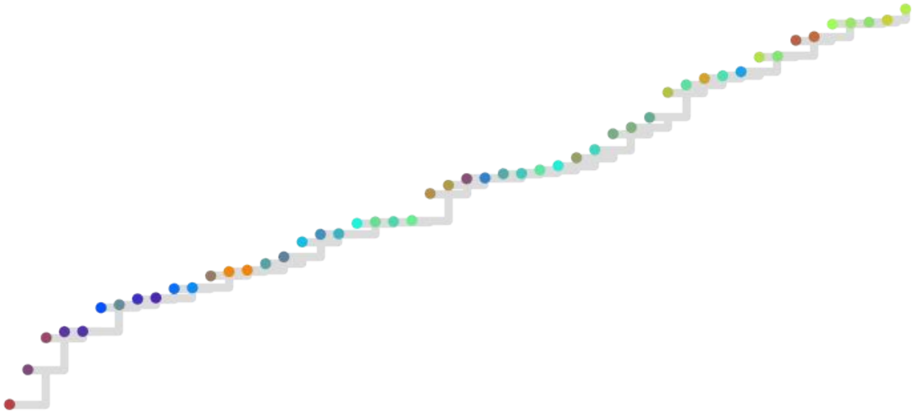
Workflow Preliminary Results



migration
bias

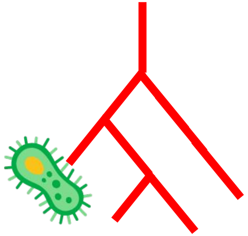




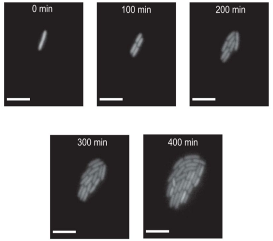


Conclusion

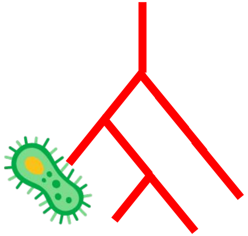
“perfect” observability



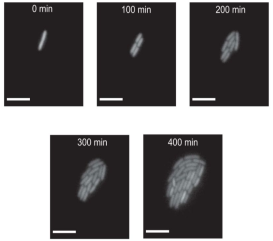
(Nozoe et al, 2017)

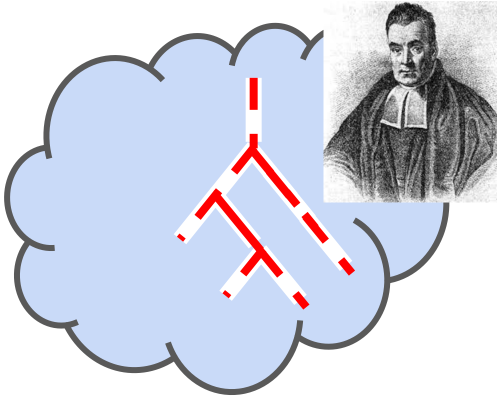


“perfect” observability



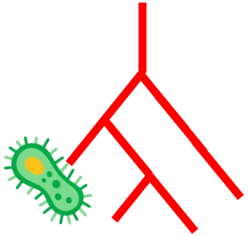
(Nozoe et al, 2017)



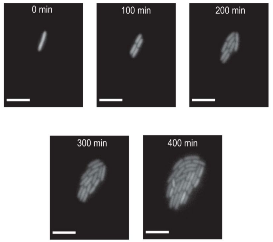


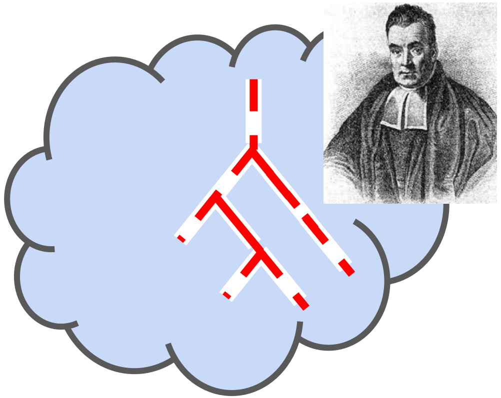
“sampling-based” observability

“perfect” observability



(Nozoe et al, 2017)



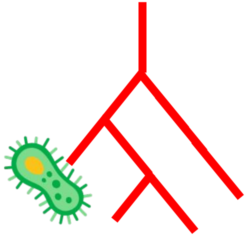


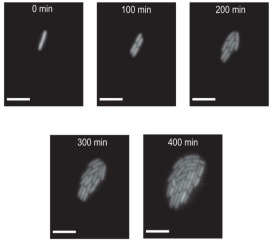
“sampling-based” observability

in vivo

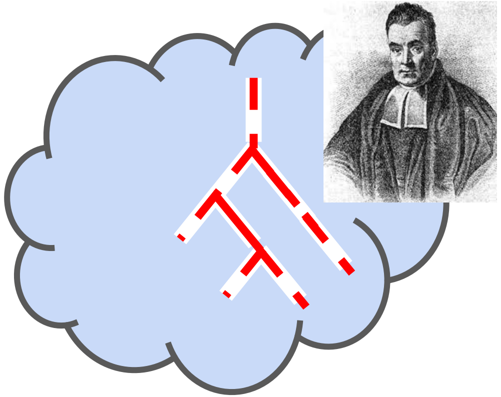
in silico

“perfect” observability





(Nozoe et al,
2017)

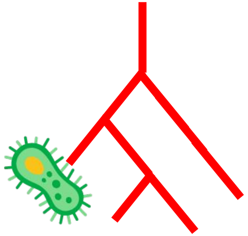


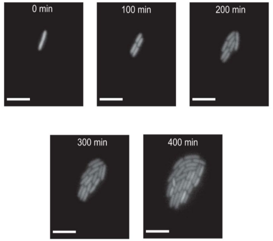
“sampling-based” observability

in vivo

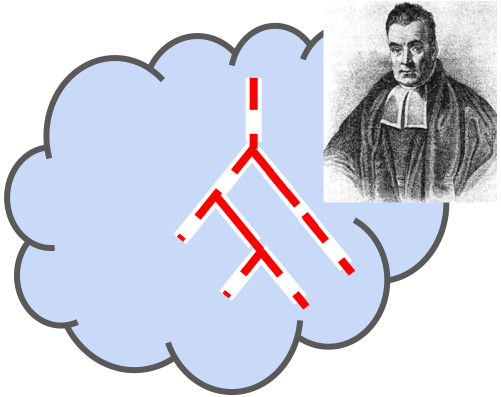
in silico

“perfect” observability





(Nozoe et al, 2017)

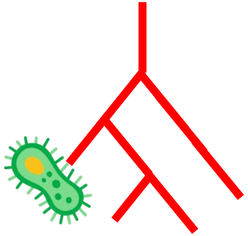


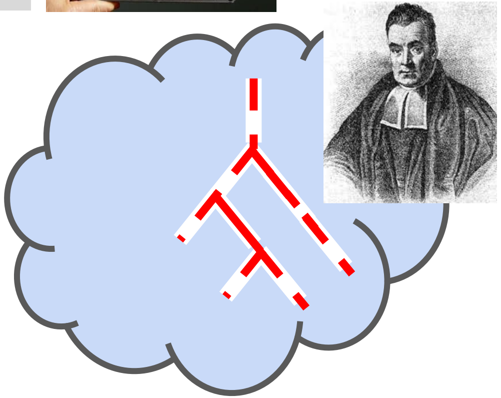
“sampling-based” observability

… ← *in vivo*

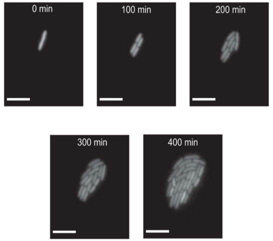
in silico → ...

“perfect” observability





(Nozoe et al, 2017)



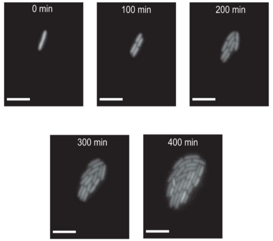
“sampling-based” observability

... ← *in vivo*

in silico → ...

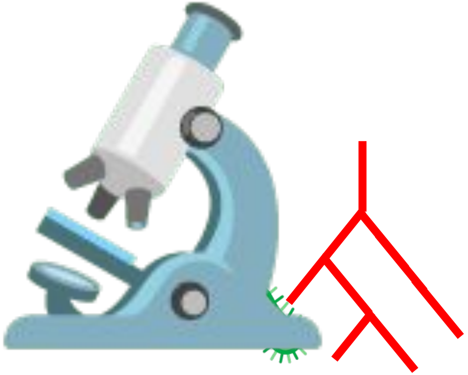
“perfect” observability

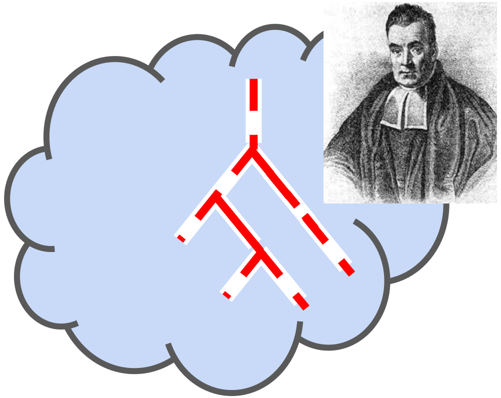
(Nozoe et al, 2017)



“sampling-based” observability

... ← *in vivo*



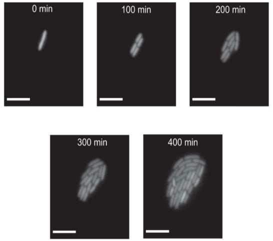


“perfect” observability

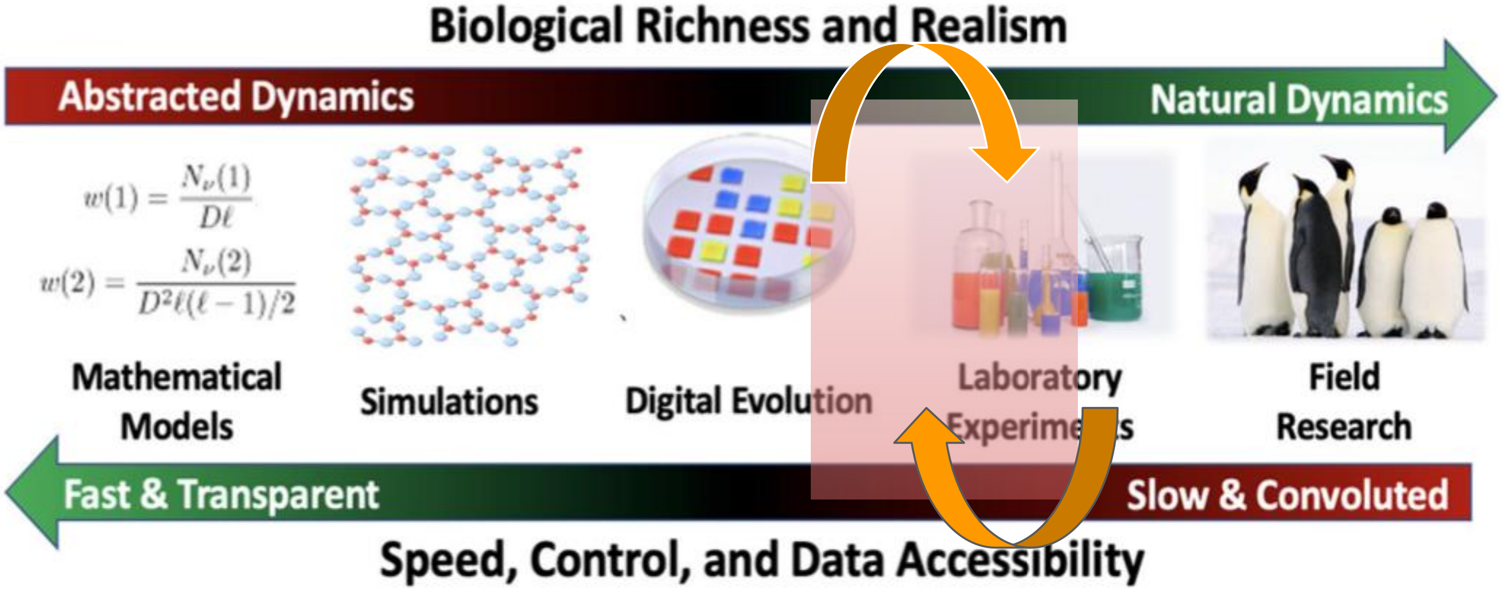
“approximate” observability

in silico → …

(Nozoe et al, 2017)



… ← *in vivo*



Biological Richness and Realism

Abstracted Dynamics

Natural Dynamics

$$w(1) = \frac{N_\nu(1)}{D\ell}$$

$$w(2) = \frac{N_\nu(2)}{D^2\ell(\ell-1)/2}$$

Mathematical Models

Simulations

Digital Evolution

Laboratory Experiments

Field Research

Fast & Transparent

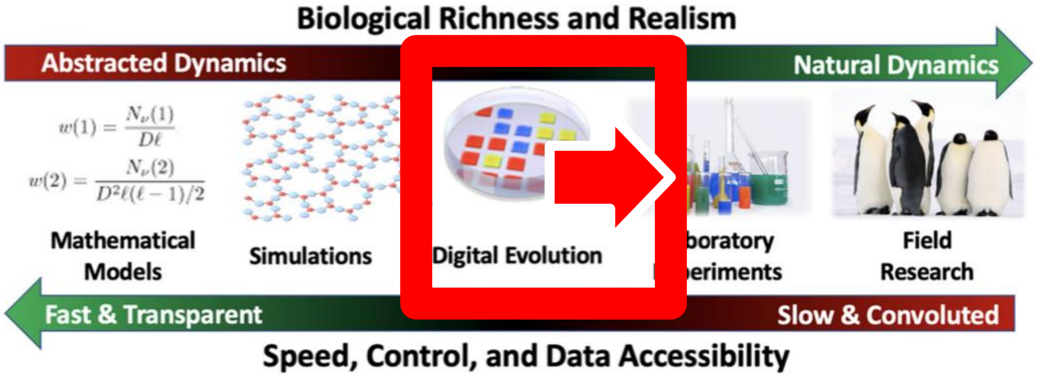
Slow & Convoluted

Speed, Control, and Data Accessibility

(Dolson and Ofria, 2021)

Parallel/distributed Scale-up: Interesting Trade-offs

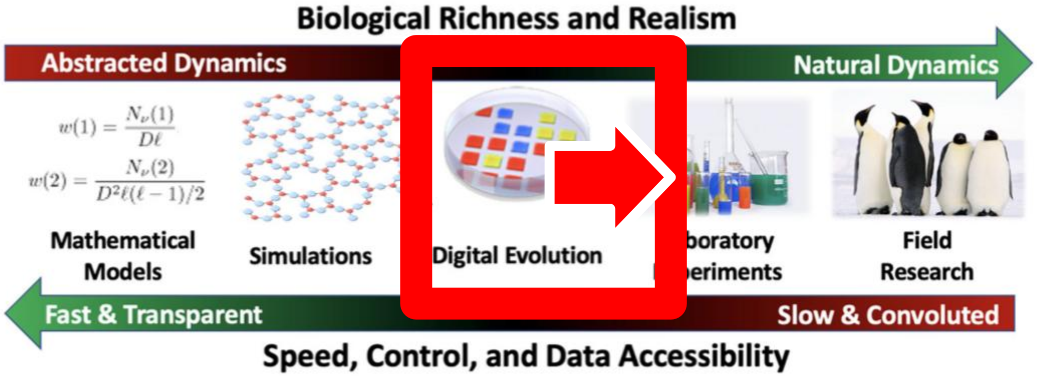
- **complete observability / sampling-based observability**
- perfect reproducibility / automated, well-specified protocols
- machine independence / measurable machine influence
- “real-time” agent execution affects fitness
- experiments on PC / experiments require non-trivial infrastructure



(Dolson and Ofria, 2021)

Parallel/distributed Scale-up: Interesting Trade-offs

- **complete observability / sampling-based observability**



(Dolson and Ofria, 2021)

Peter J. Park, CC BY 2.5
<https://creativecommons.org/licenses/by/2.5>, via
Wikimedia Commons
kqedquest Via Flickr

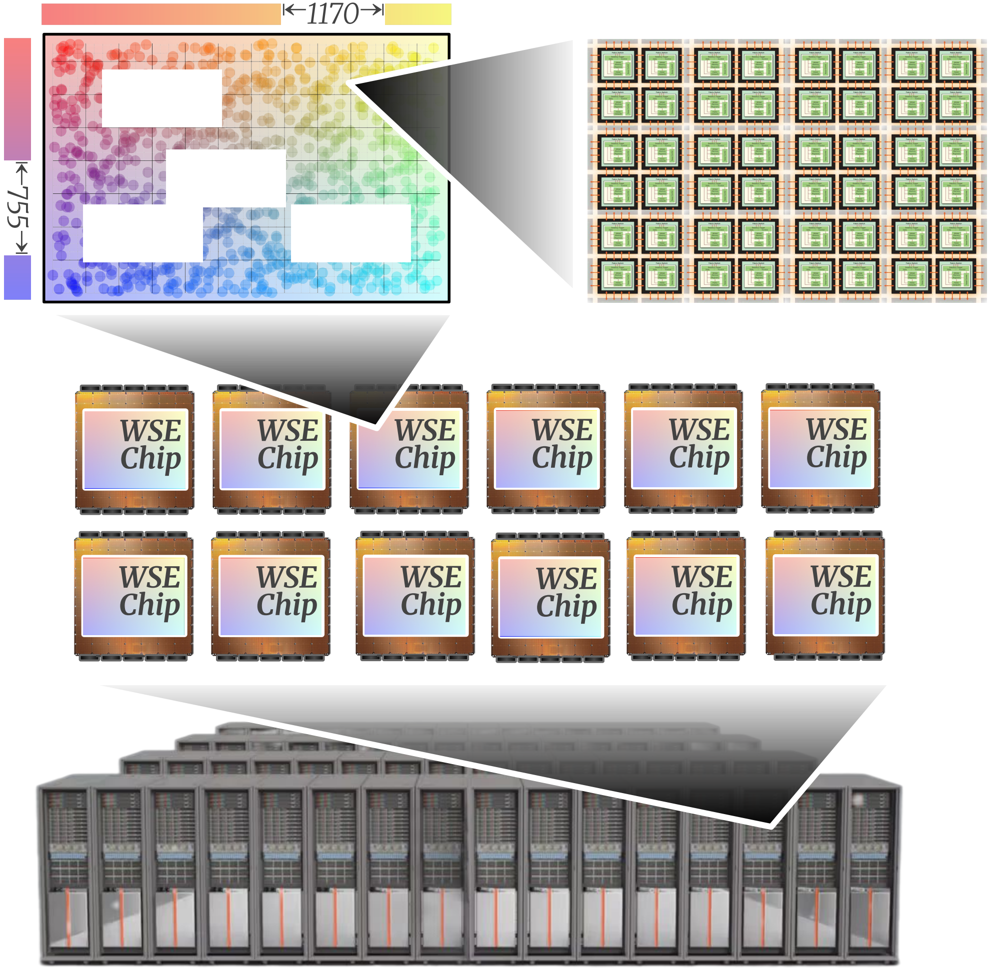
Wafer-Scale Cluster

Peter J. Park, CC BY 2.5
<https://creativecommons.org/licenses/by/2.5>, via
Wikimedia Commons
kqedquest Via Flickr

2,048
CS-3

Wafer-Scale Cluster

Peter J. Park, CC BY 2.5
<https://creativecommons.org/licenses/by/2.5>, via
Wikimedia Commons
kqedquest Via Flickr



1.74 B
processors

2,048
CS-3

256
exaFLOPs

Wafer-Scale Cluster

Today

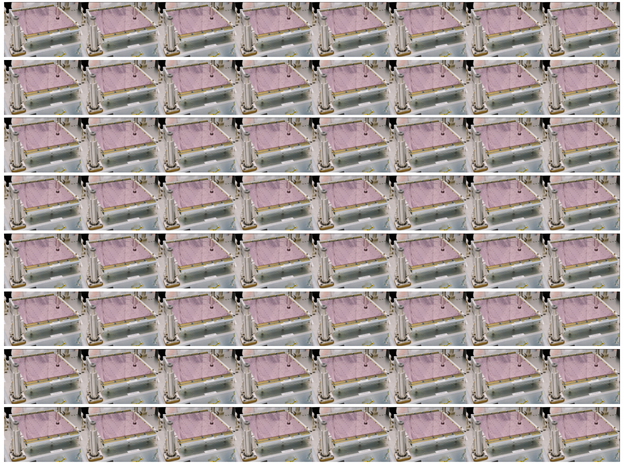
PSC Neocortex
(NSF ACCESS)

Argonne Leadership Compute Facility
(Dept of Energy)

Soon

Condor Galaxy 3
(Upcoming)

64 CS-3 chips (54 million cores)



Today

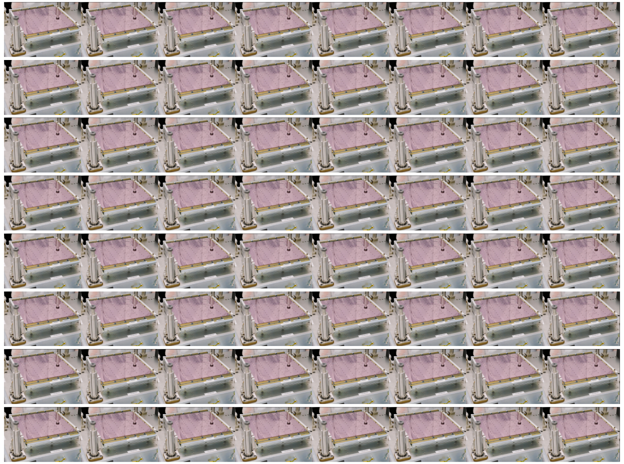
PSC
Neocortex
(NSF ACCESS)

Argonne
Leadership
Compute Facility
(Dept of Energy)

Soon

Condor Galaxy 3
(Upcoming)

64 CS-3 chips (54 million cores)



Future

???

As many as 2,048 systems can be combined…

1.84 billion cores

https://spectrum.ieee.org/cerebras-chip-cs3

ByteBoost
Workshop '24

PSC

NEOCORTEX


Dr. Luis Zaman


Dr. Emily Dolson


Joey Wagner


Connor Yang (UROP)


Vivaan Singhvi (UROP)

COMPLEX SYSTEMS

EEB

MIDAS

U.S. DEPARTMENT OF ENERGY | Office of Science

Office of Advanced Scientific Computing Research (ASCR)
Award Number DE-SC0025634

This project is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Sciences program.

🐦@MorenoMatthewA 🐘 @mas.to

📧 morenoma@umich.edu

ByteBoost Workshop ‘24

“Re: Re: Fwd: Troubleshooting…”

From Sylvia Howland

To Paola Buitrago

Matthew Andres Moreno
morenoma@umich.edu

Leighton Wilson

Mathias Jacquelin

Cc PSC Neocortex (Mei Yu, Julian, & Riaz)

Paola Buitrago

Date Feb 2, 2026 at 12:10

Standard encryption (TLS)
Learn more

(Extensive)
Technical Assistance

🐦@MorenoMatthewA 🐘 @mas.to

📧 morenoma@umich.edu

these slides: hopth.ru/gg

Questions?

We have a very bad tendency to base our plans for computers on the equipment we have in house and the things we're doing now. And totally fail to review them in the light of the equipment that will be available and the things that we will be doing — I think the saddest phrase I ever hear in a computer installation is that horrible one "but we've always done it that way." That's a forbidden phrase in my office.

Capt. Grace Hopper

Future Possibilities: Data, Hardware, Software, and People — August 26, 1982

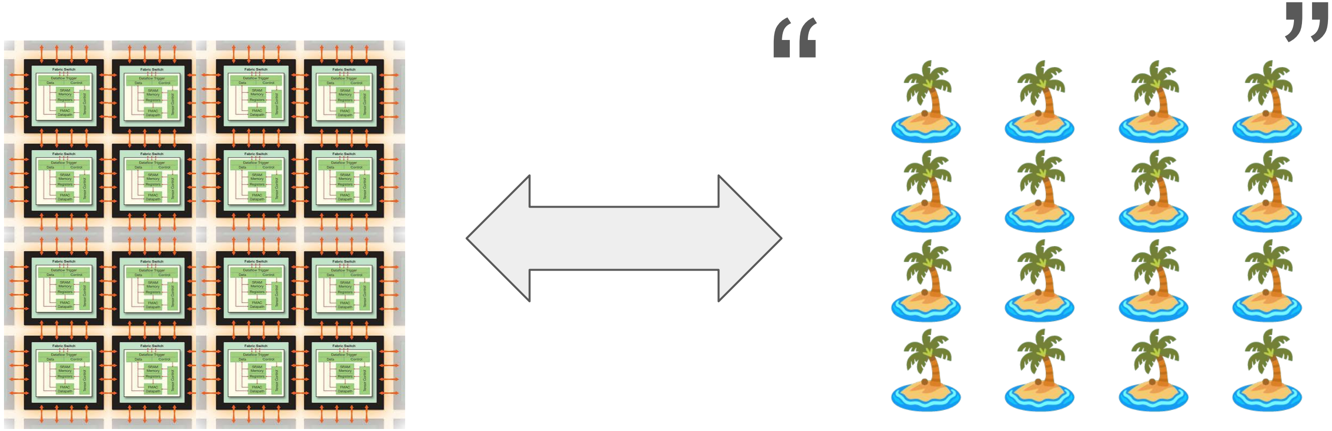
References

- Ackley, David H. "A Robust Programmable Replicator for an Indefinitely Scalable Machine." ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference. MIT Press, 2023.
- Liquidware. Introducing the Illuminato X Machina. http://antipastohw.blogspot.com/2009/08/introducing-illuminato-x-machina.html
- ALIEN Artificial Life Environment. https://www.alien-project.org/index.html
- Dolson, Emily, and Charles Ofria. "Digital evolution for ecology research: a review." Frontiers in Ecology and Evolution 9 (2021): 750779.
- Good, Benjamin H., et al. "The dynamics of molecular evolution over 60,000 generations." Nature 551.7678 (2017): 45-50.
- Ofria, Charles, and Claus O. Wilke. "Avida: A software platform for research in computational evolutionary biology." Artificial life 10.2 (2004): 191-229.
- Buitrago P.A., Nystrom N.A. (2021) Neocortex and Bridges-2: A High Performance AI+HPC Ecosystem for Science, Discovery, and Societal Good. In: Nesmachnow S., Castro H., Tchernykh A. (eds) High Performance Computing. CARLA 2020. Communications in Computer and Information Science, vol 1327. Springer, Cham. https://doi.org/10.1007/978-3-030-68035-0_15

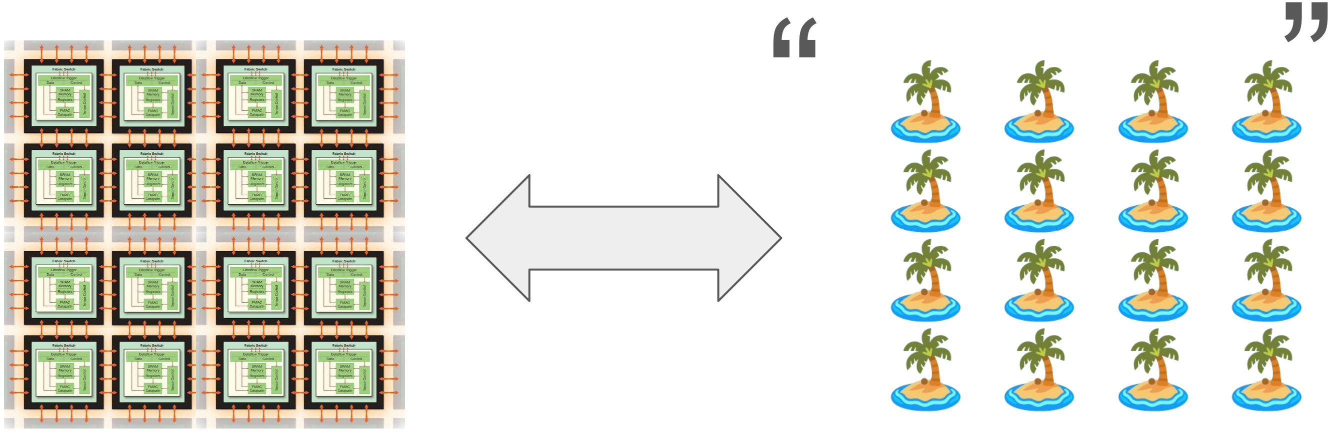
Problem Setup — [props?]

Problem Setup — [props?]

Proof-of-concept Experiment



Proof-of-concept Experiment



simple agent model, explicit fitness

a) purifying regime

👾 *=mutate⇒*

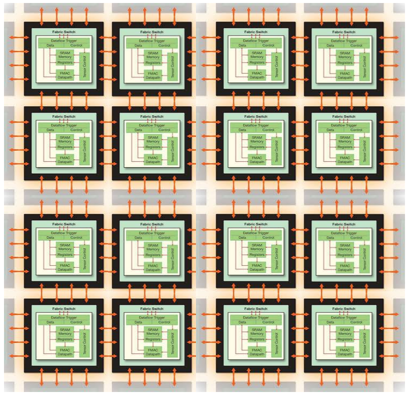
👾 or 👾$^-$

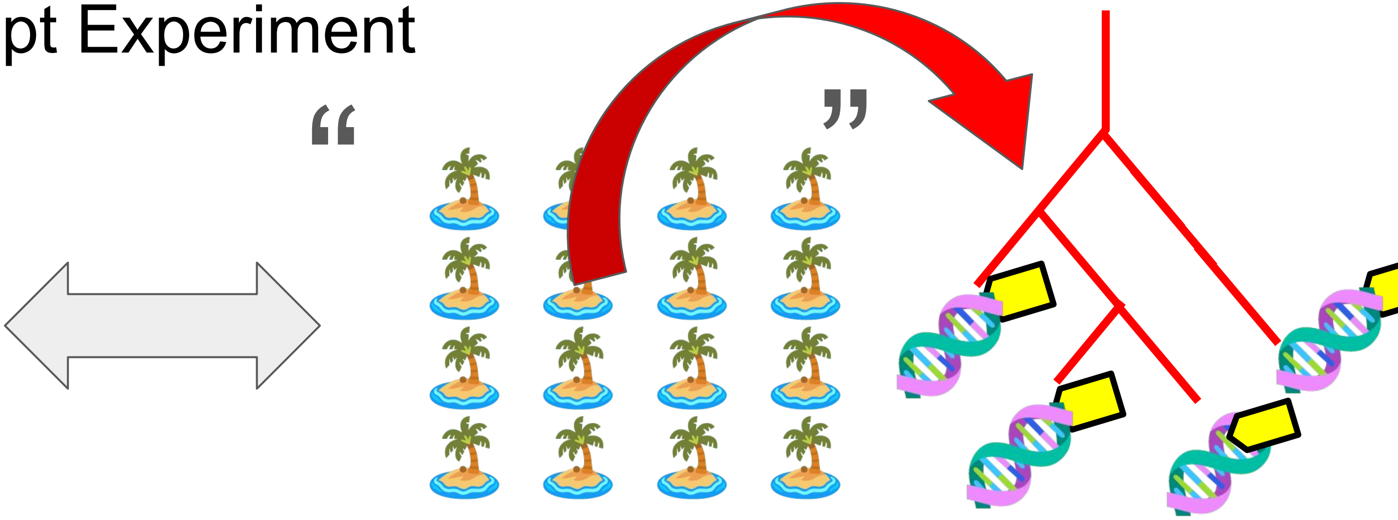
b) adaptive regime

👾 *=mutate⇒*

👾 or 👾$^-$ or 👾$^+$

Proof-of-concept Experiment





simple agent model, explicit fitness

a) purifying regime

👾 *=mutate⇒*

👾 or 👾—

b) adaptive regime

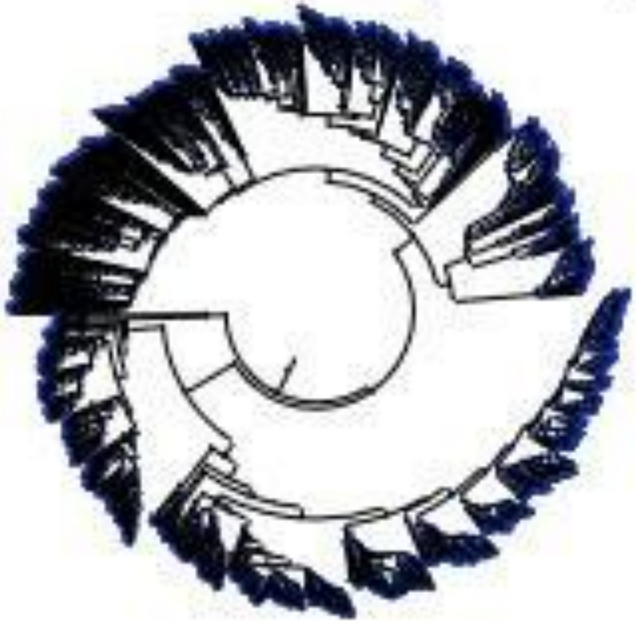
👾 *=mutate⇒*

👾 or 👾— or 👾+

Example Phylogenies

~ 8 million agents
~ 10k tip phylogenies

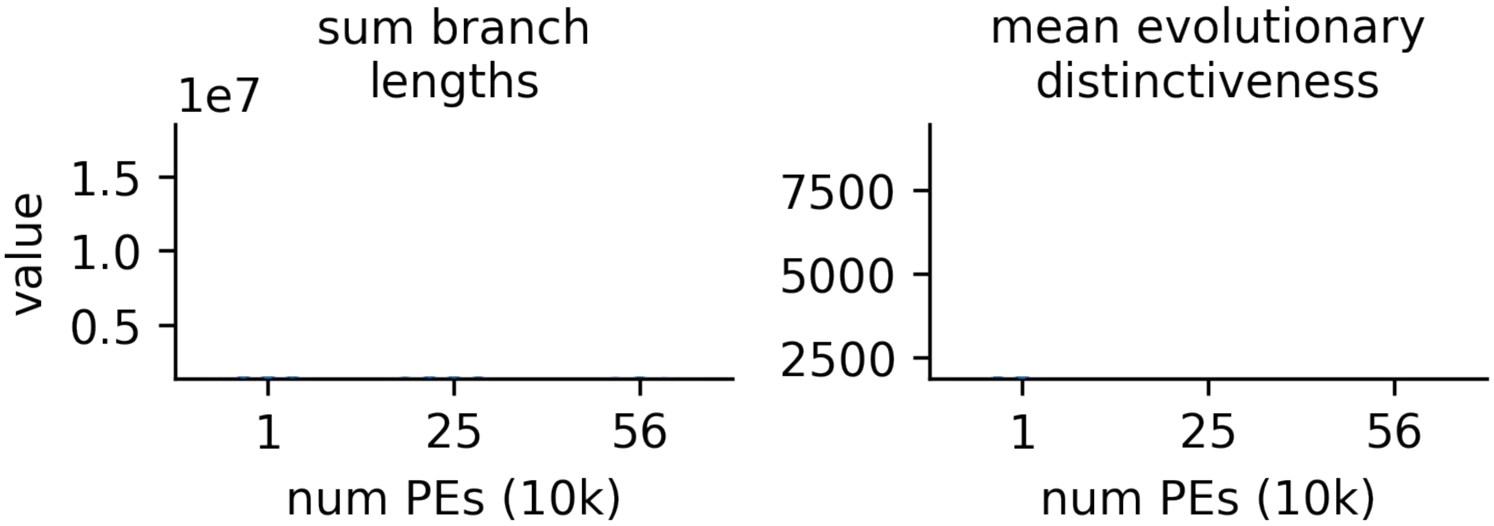
a) purifying regime



b) adaptive regime

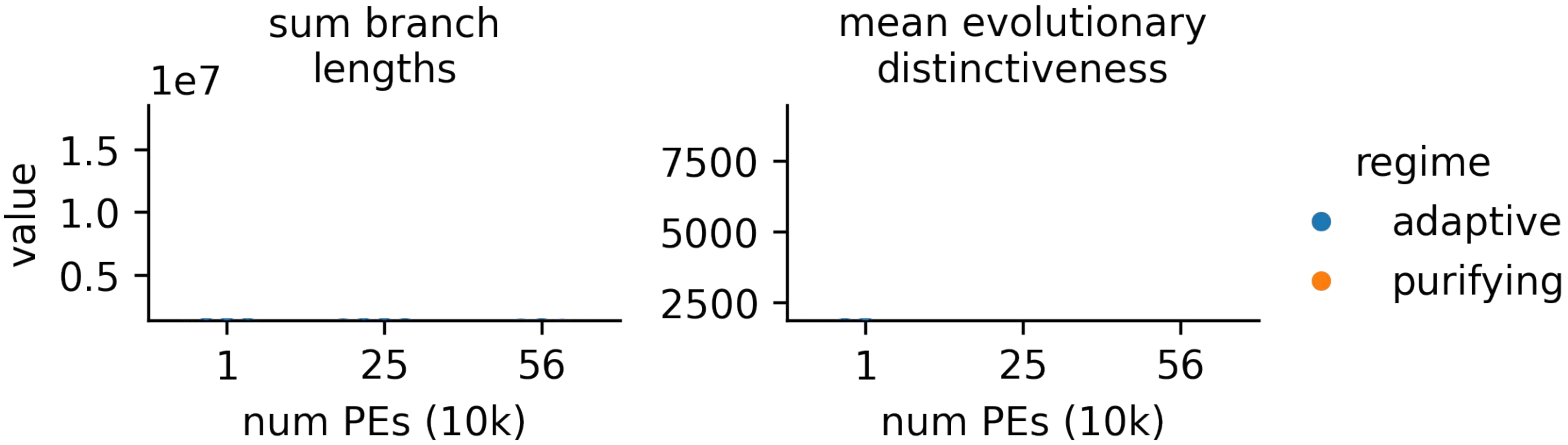
morenoma@umich.edu

Phylogeny Structure Metrics



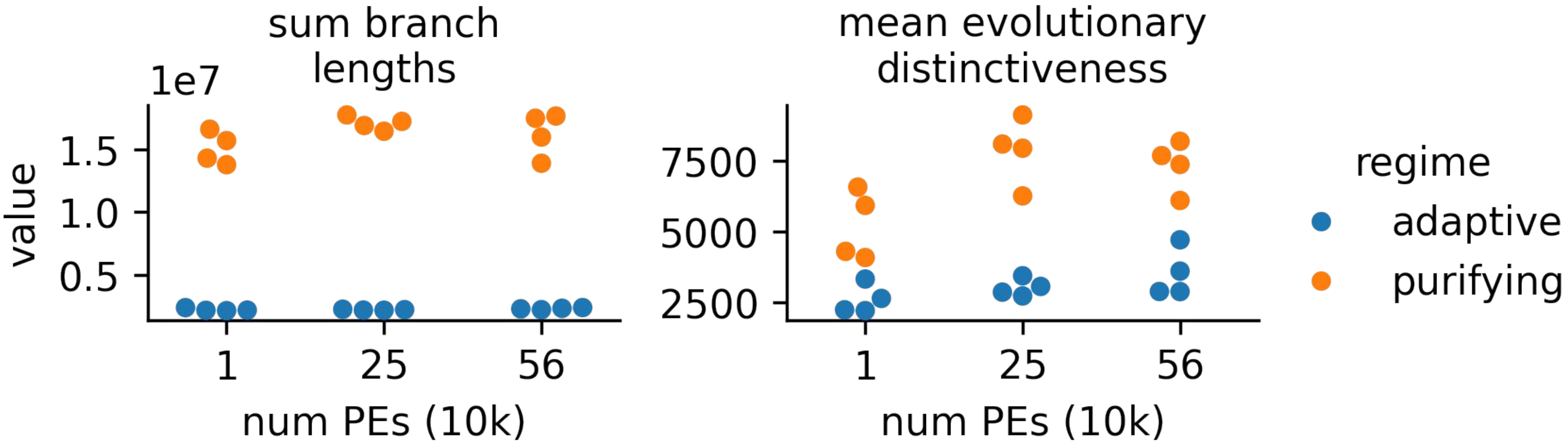
📚 [slides, paper, code] https://hopth.ru/de

Phylogeny Structure Metrics



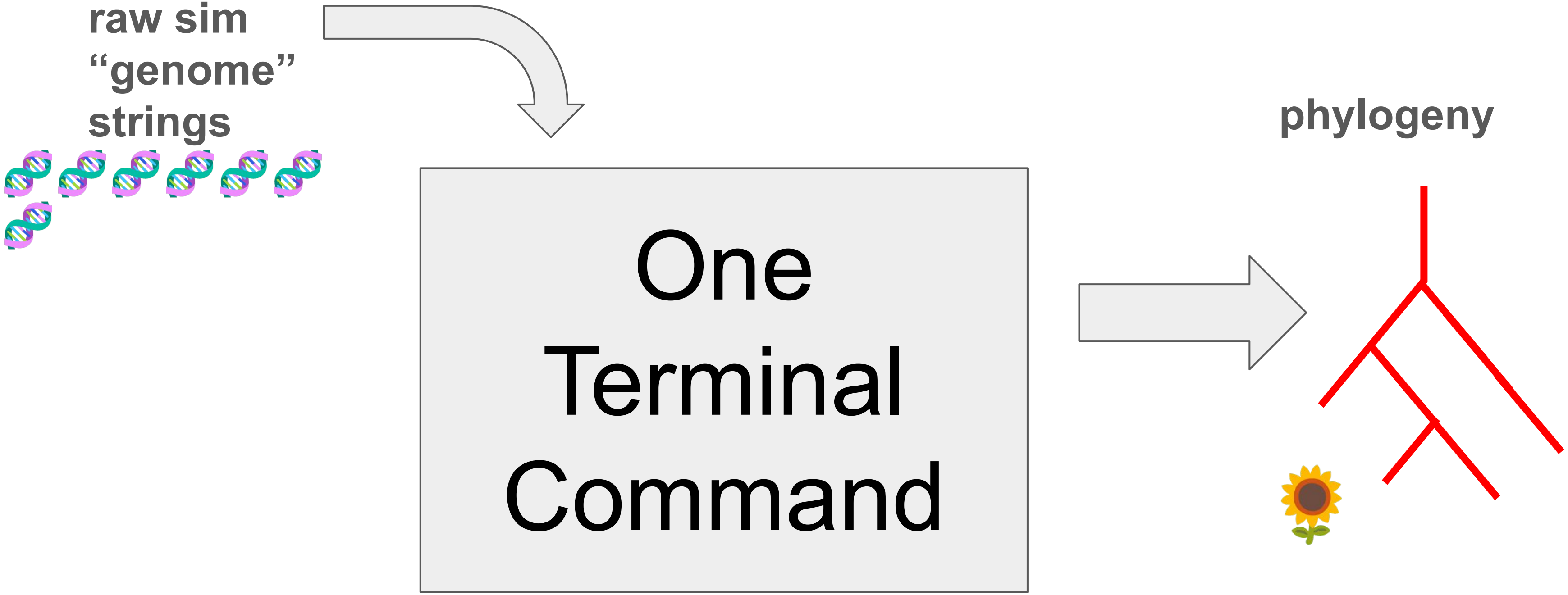
📚 [slides, paper, code] https://hopth.ru/de

Phylogeny Structure Metrics



📚 [slides, paper, code] https://hopth.ru/de

Goal: end-to-end encapsulated workflow



raw sim “genome” strings

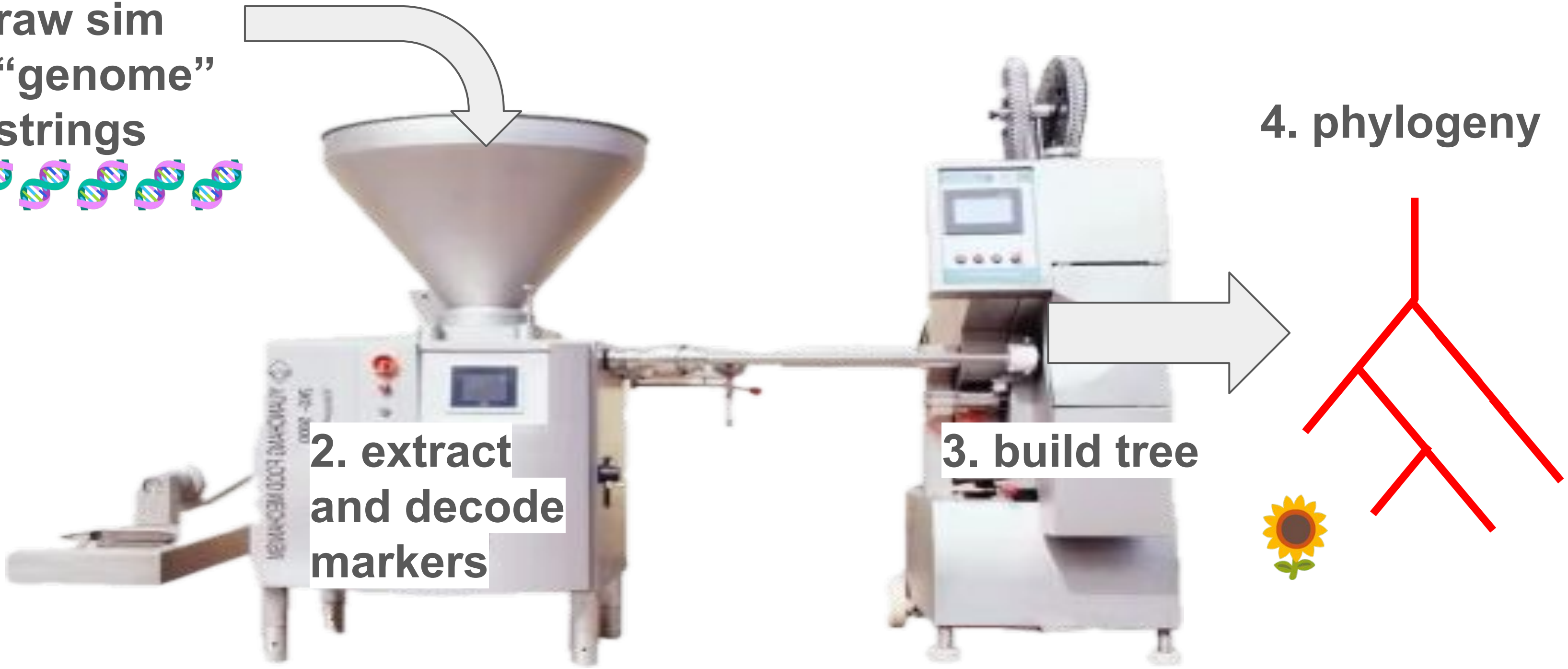
🧬🧬🧬🧬🧬🧬🧬

One Terminal Command

phylogeny

🌻

Goal: end-to-end encapsulated workflow



1. **raw sim "genome" strings**

🧬🧬🧬🧬🧬🧬🧬

2. extract and decode markers

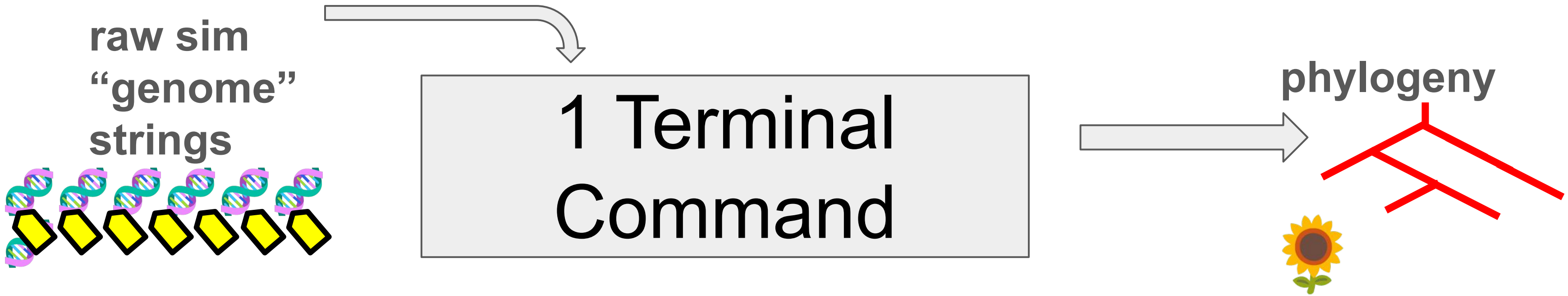
3. build tree

4. phylogeny

🌻

courtesy meat-machinery.com

package



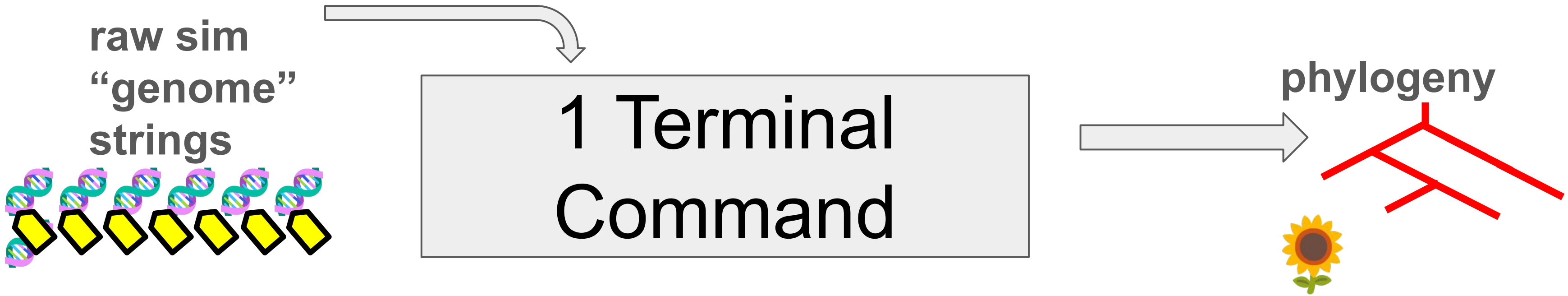
raw sim
“genome”
strings

1 Terminal Command

phylogeny

Goal: high-performance, easy workflow

package


raw sim “genome” strings

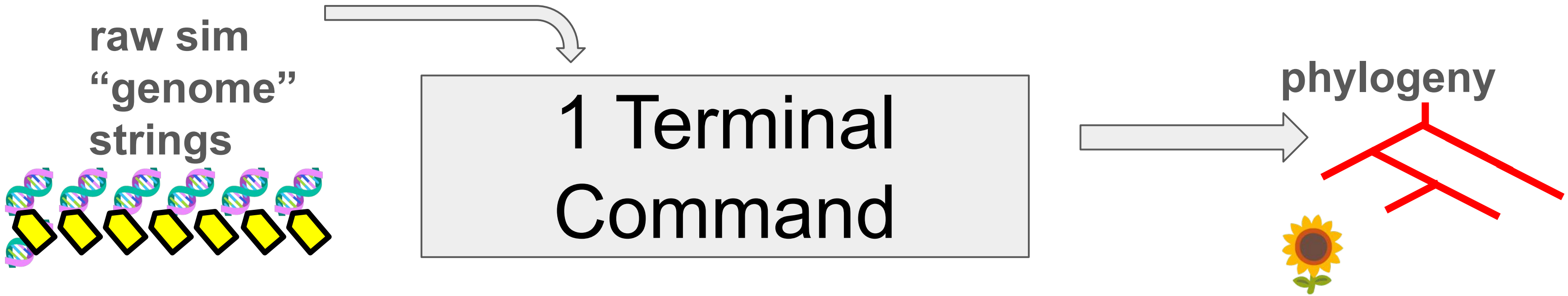
1 Terminal Command

phylogeny

Goal: high-performance, easy workflow

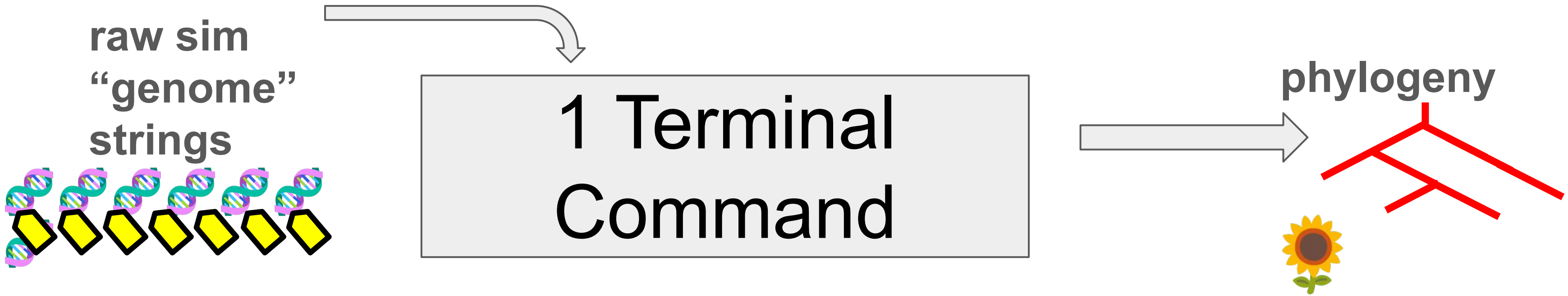
- today: 2.5 million tips per minute
- goal: >5 million tips per minute

package



Goal: high-performance, easy workflow

- today: 2.5 million tips per minute
- goal: >5 million tips per minute



raw sim
“genome”
strings

1 Terminal Command

phylogeny

Goal: high-performance, easy workflow

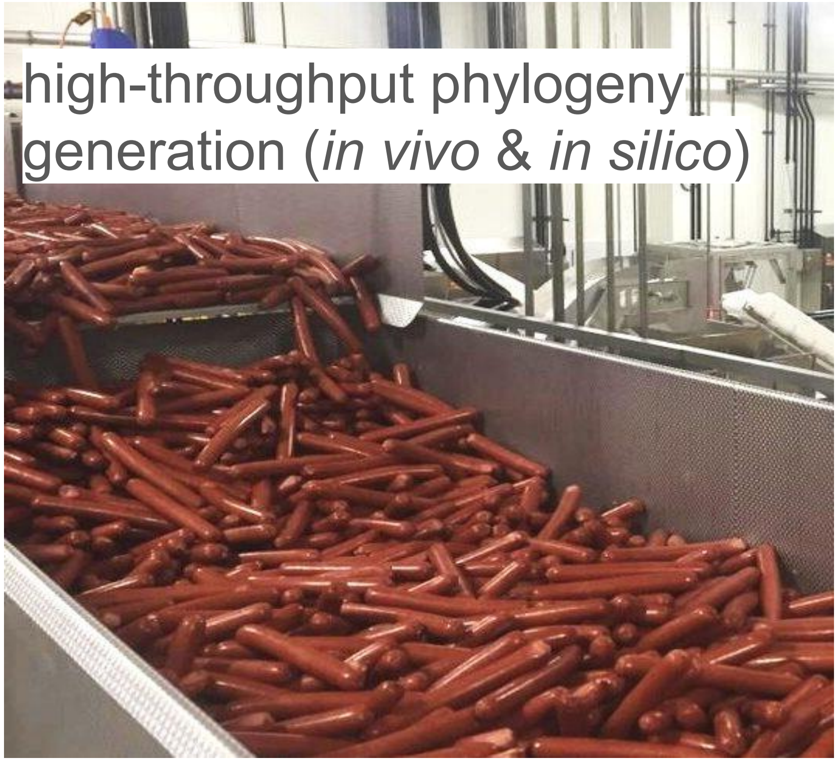
- today: 2.5 million tips per minute
- goal: >5 million tips per minute
- prototype —> python package

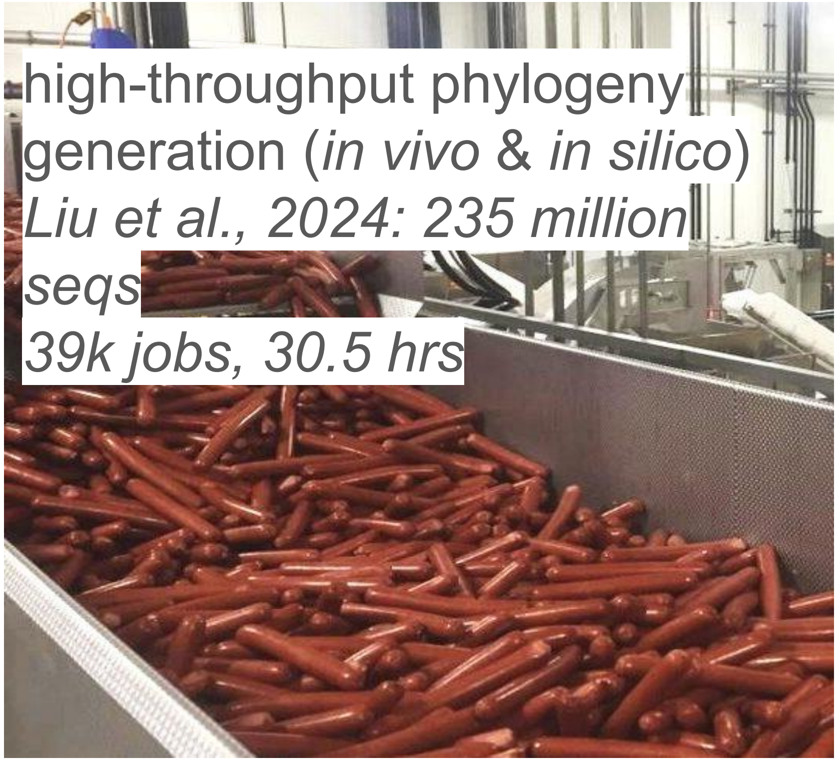
Connor Yang

Vivaan Singhvi

Joey Wagner

high-throughput phylogeny generation (*in vivo* & *in silico*)

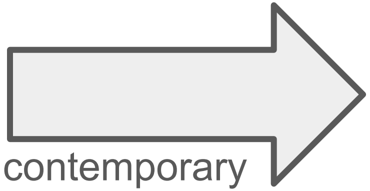




high-throughput phylogeny generation (*in vivo* & *in silico*)

Liu et al., 2024: 235 million seqs

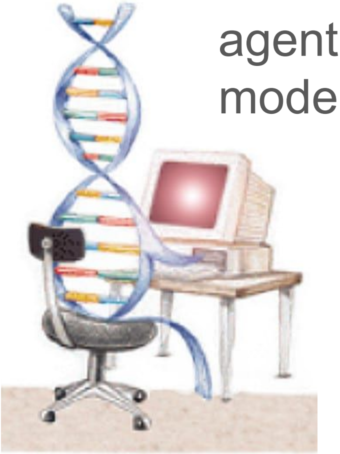
39k jobs, 30.5 hrs



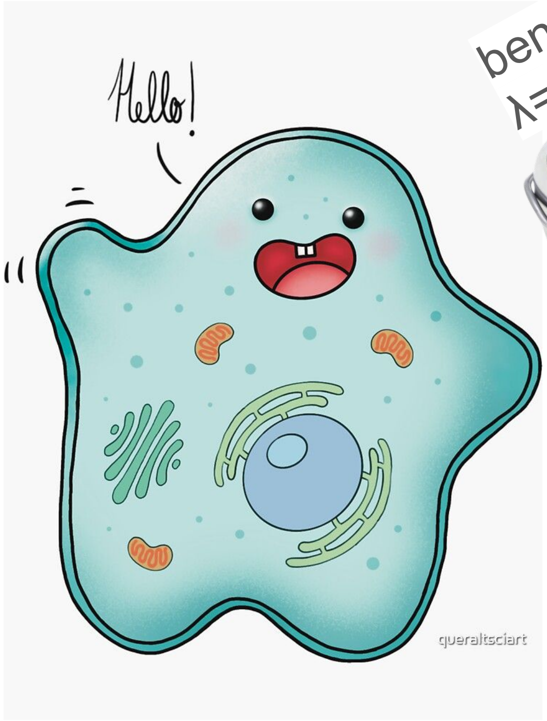
contemporary analyses

Future Work

Results



agent-based
model

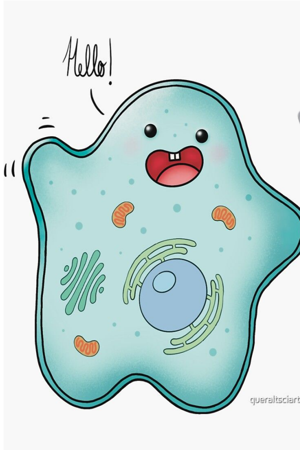


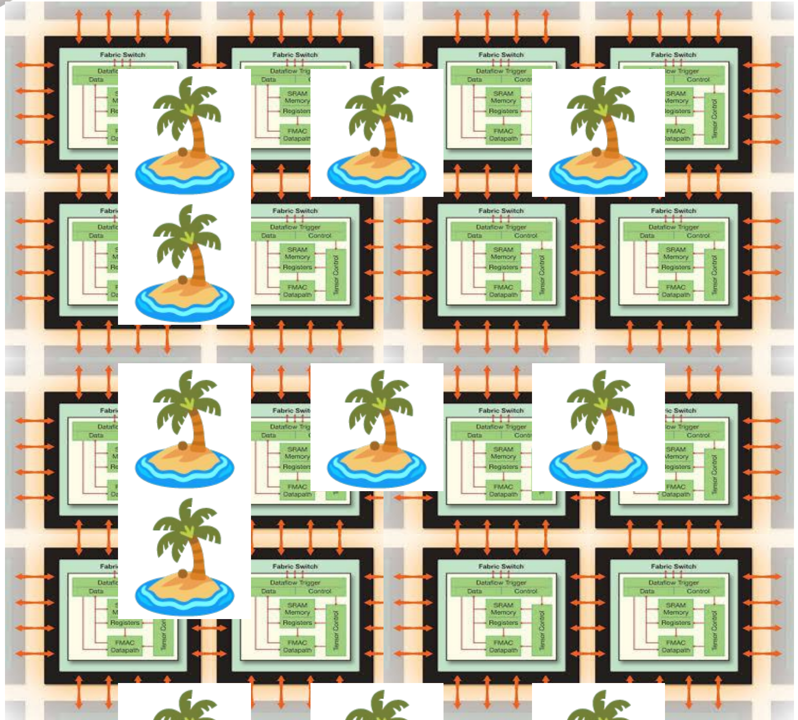
beneficial
λ=1 in 100k

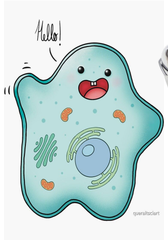
deleterious
λ=1 in 1k

poisson distribution

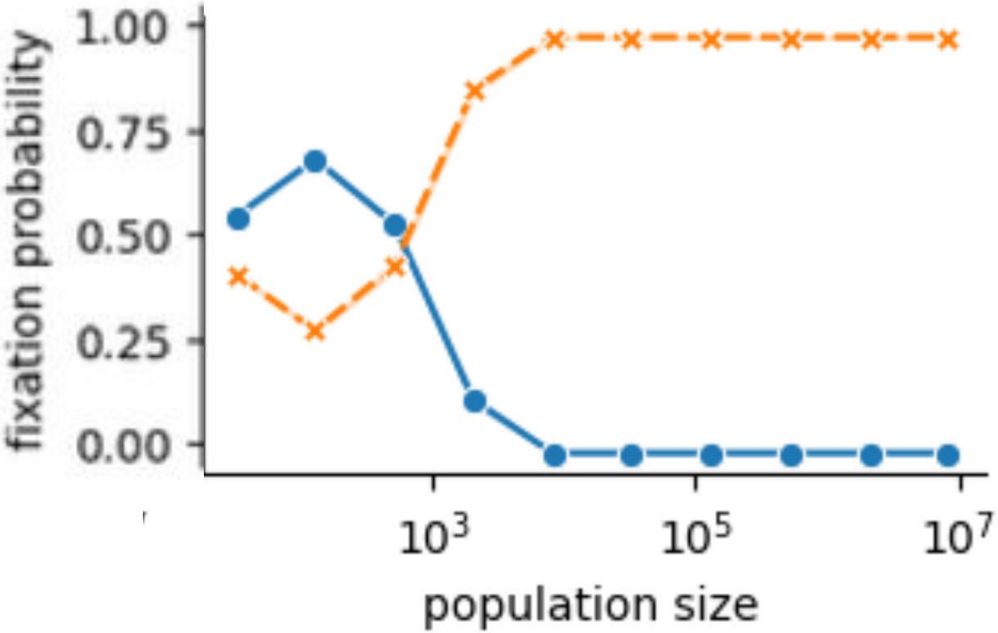
agent-based model

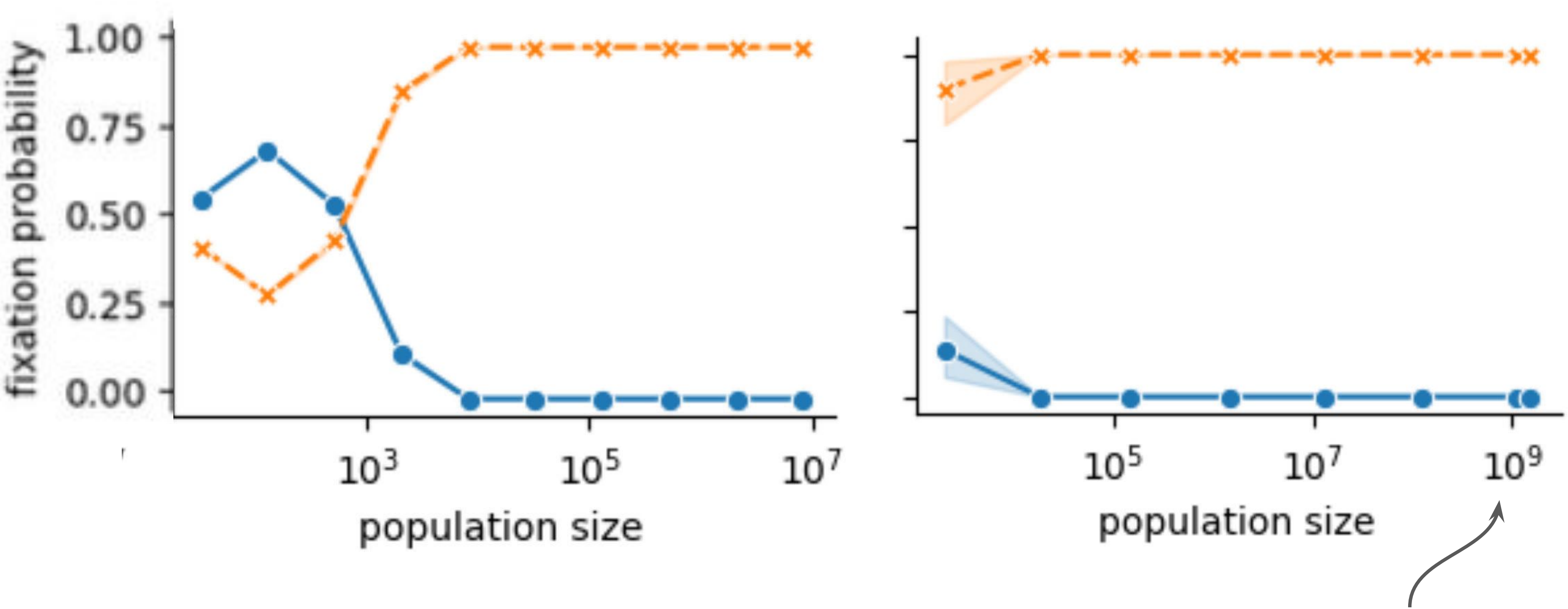




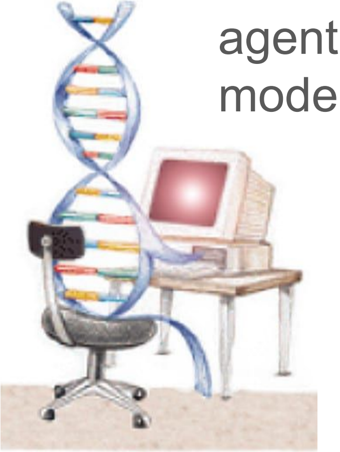

Hello!


0002
0002

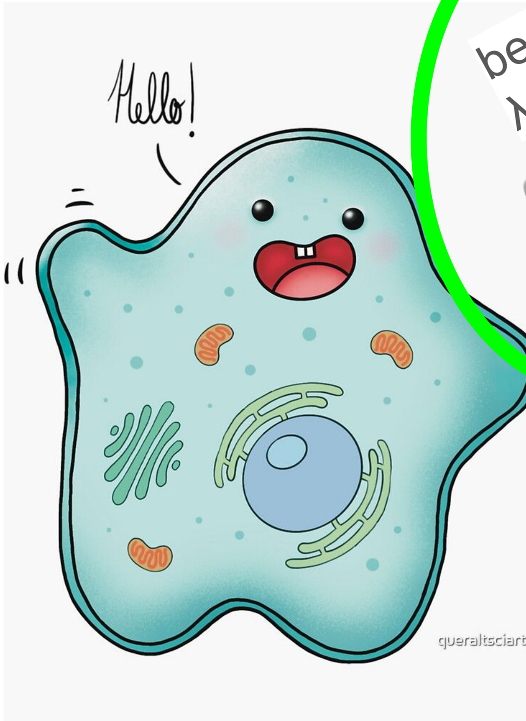

fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size



1 billion agents



agent-based
model

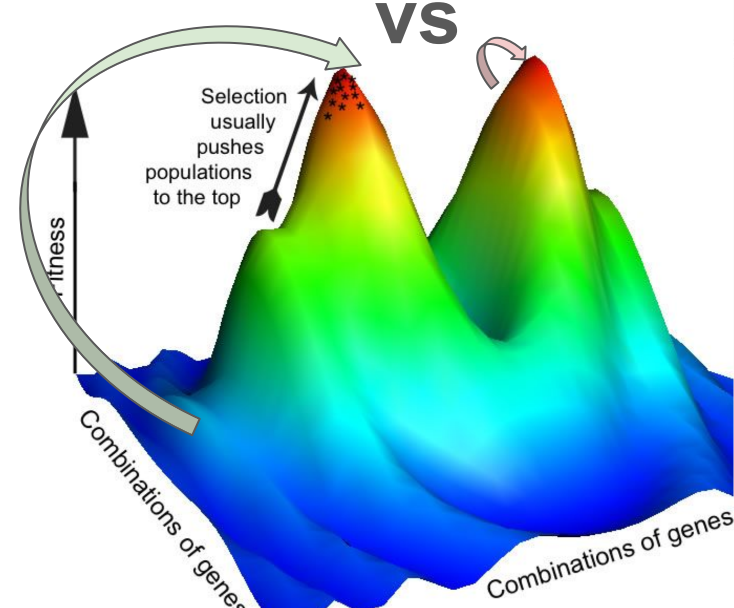


∞

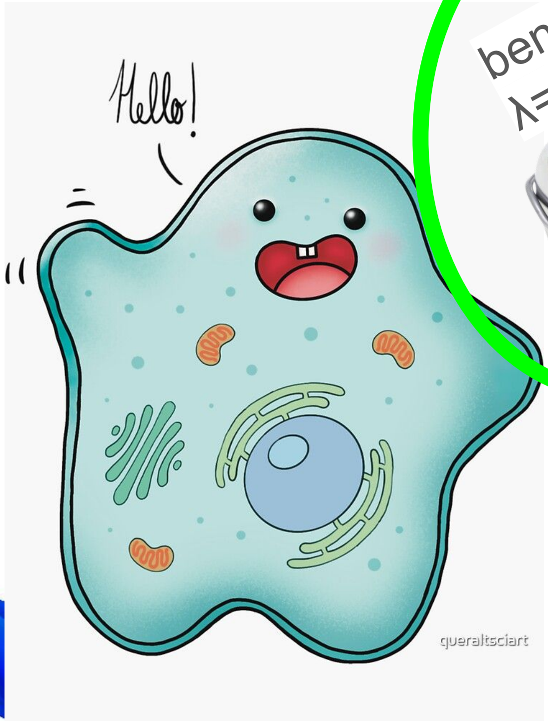
beneficial
λ=1 in 100k

deleterious
λ=1 in 1k

agent-based
model



vs



∞

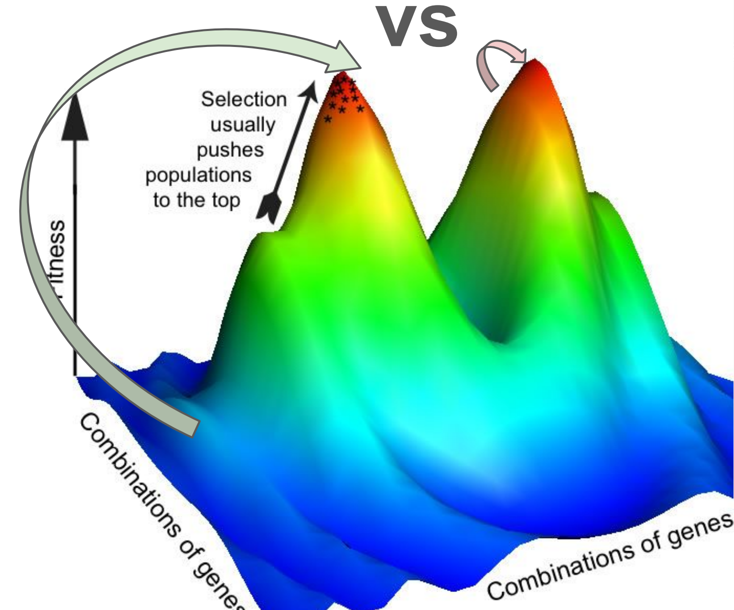
beneficial
λ=1 in 100k

deleterious
λ=1 in 1k

poisson distribution

agent-based
model

vs

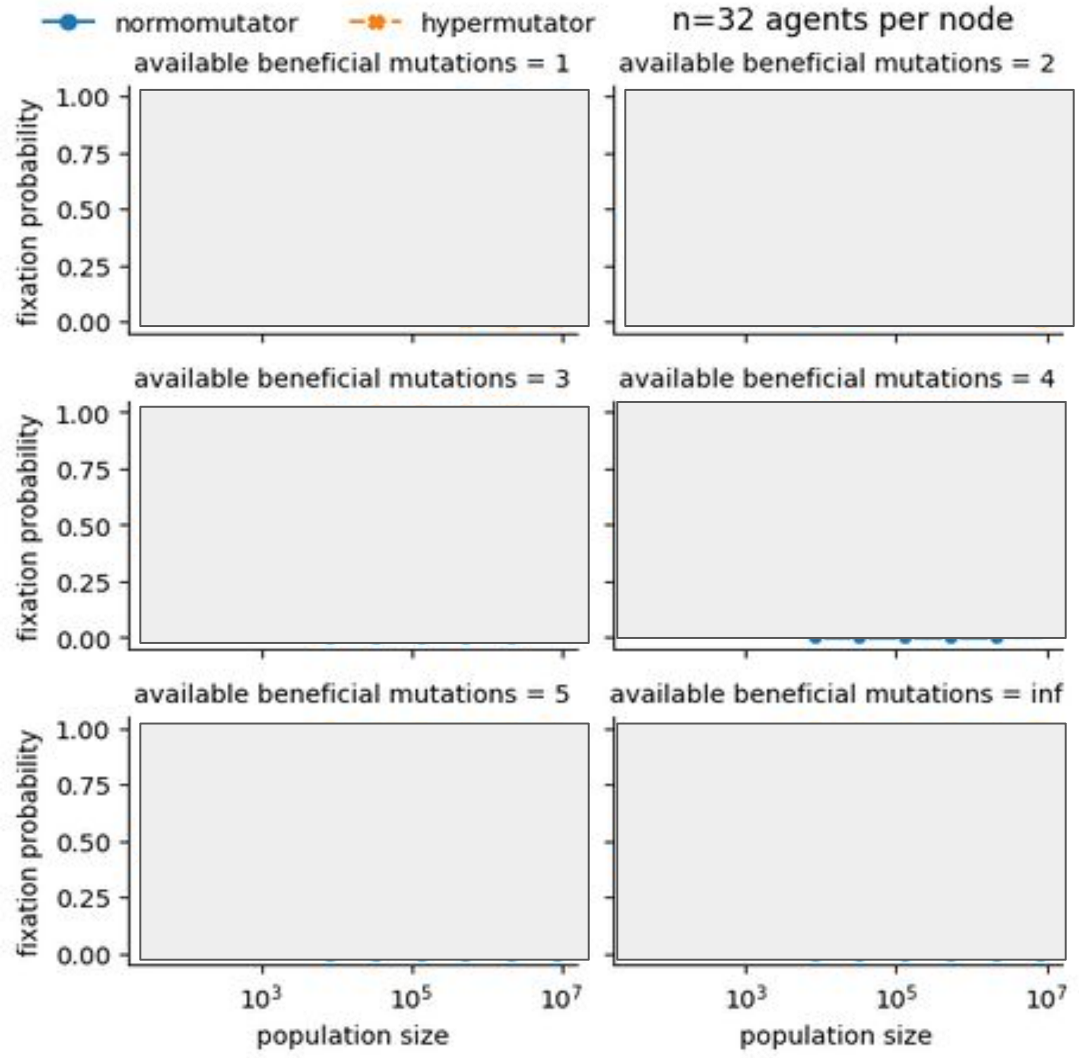


beneficial
λ=1 in 100k

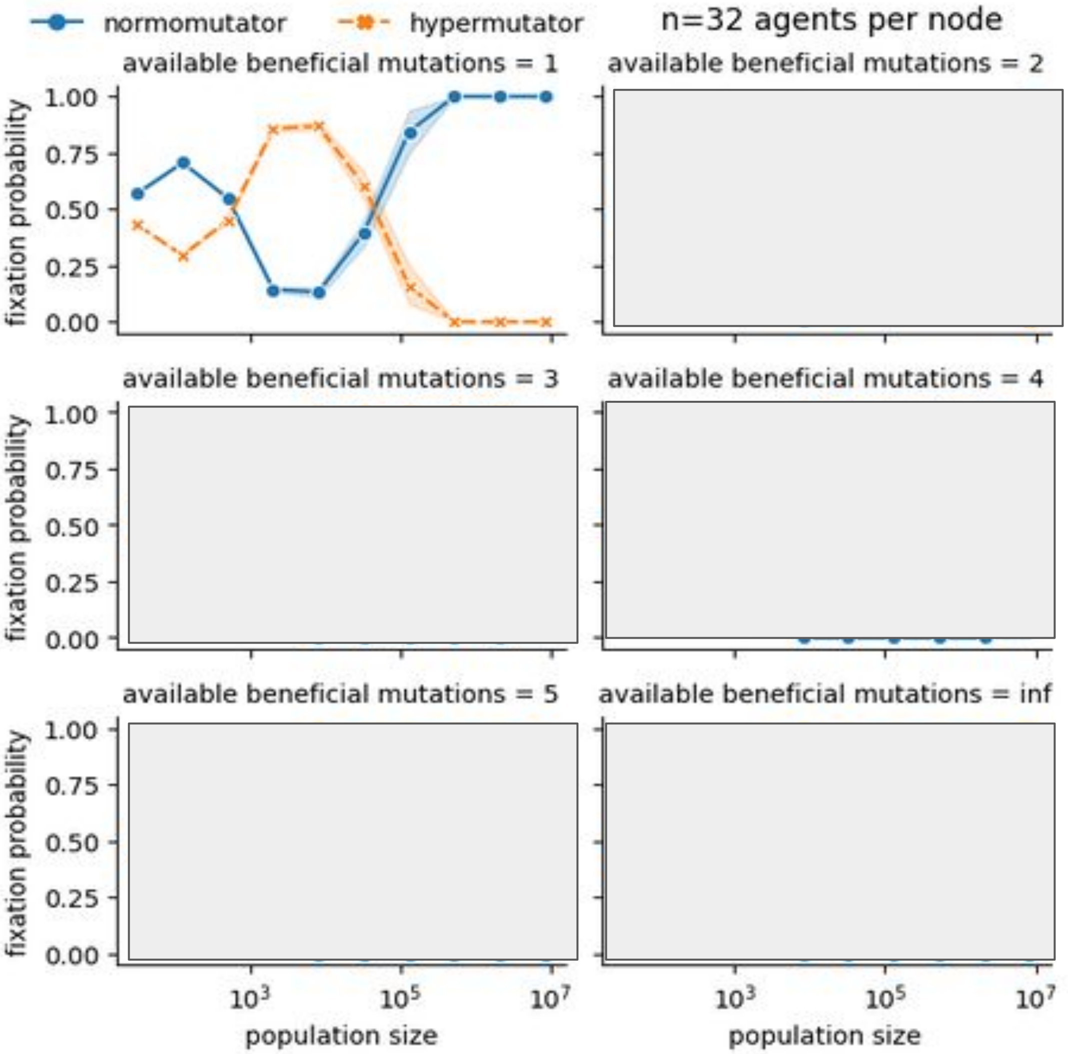
deleterious
λ=1 in 1k

max n

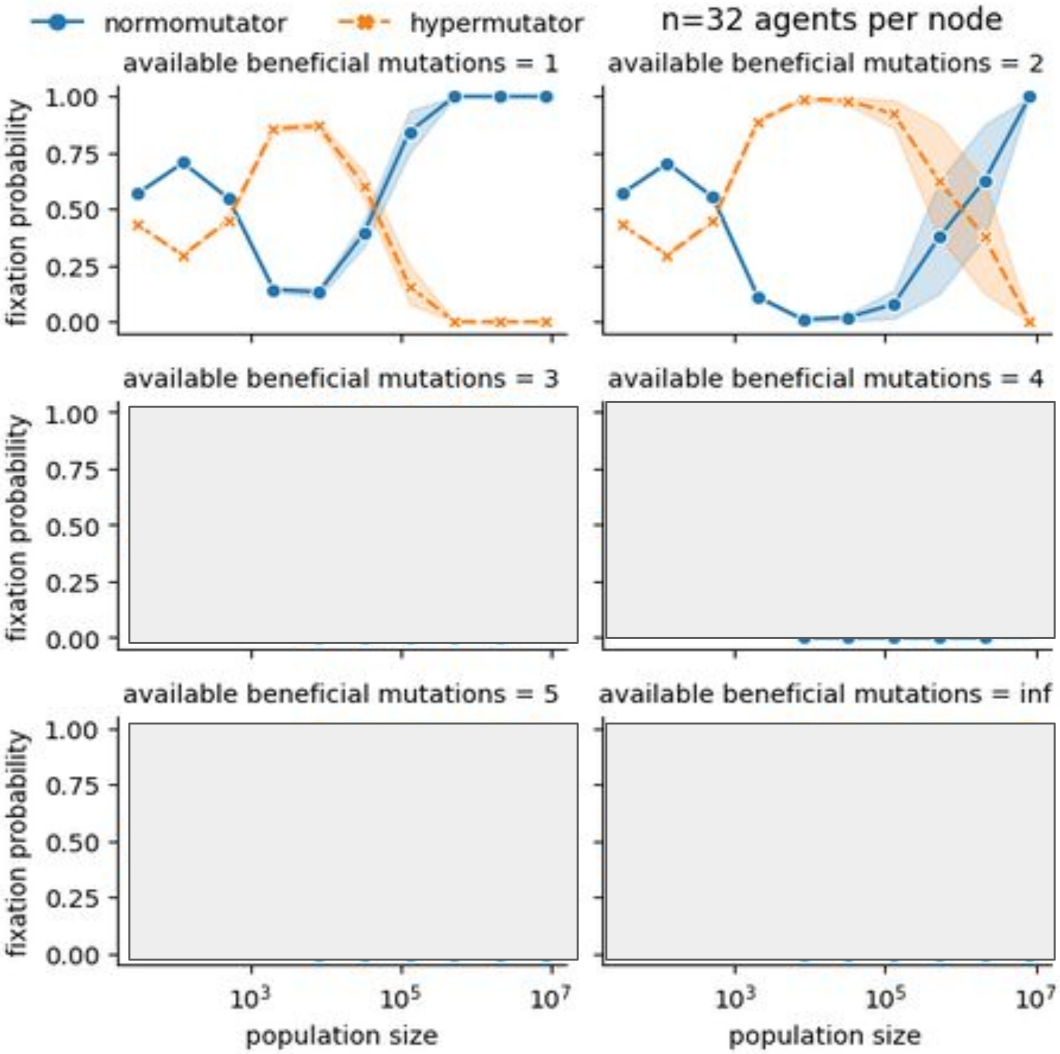
poisson distribution


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

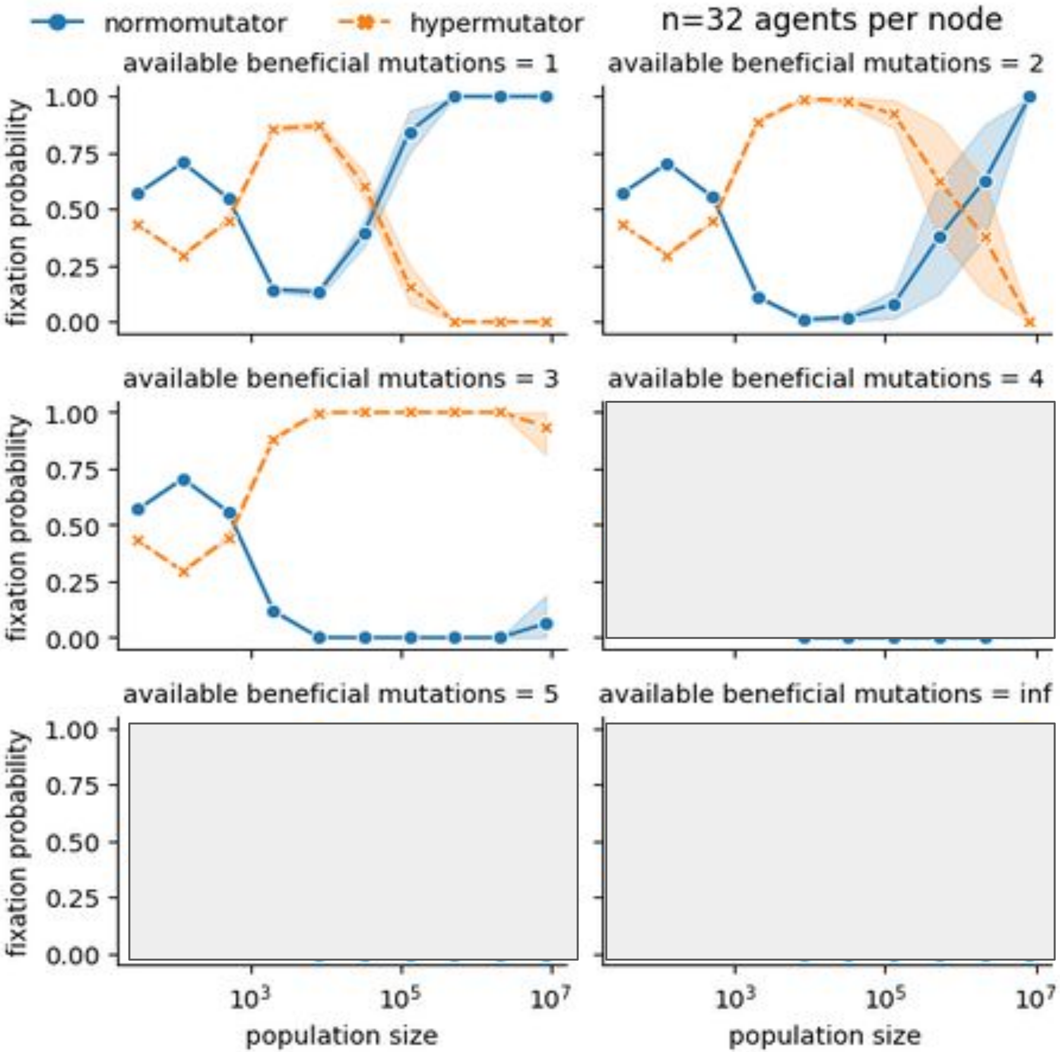

Hello!
STOP
0002
0002
queraltsciart


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

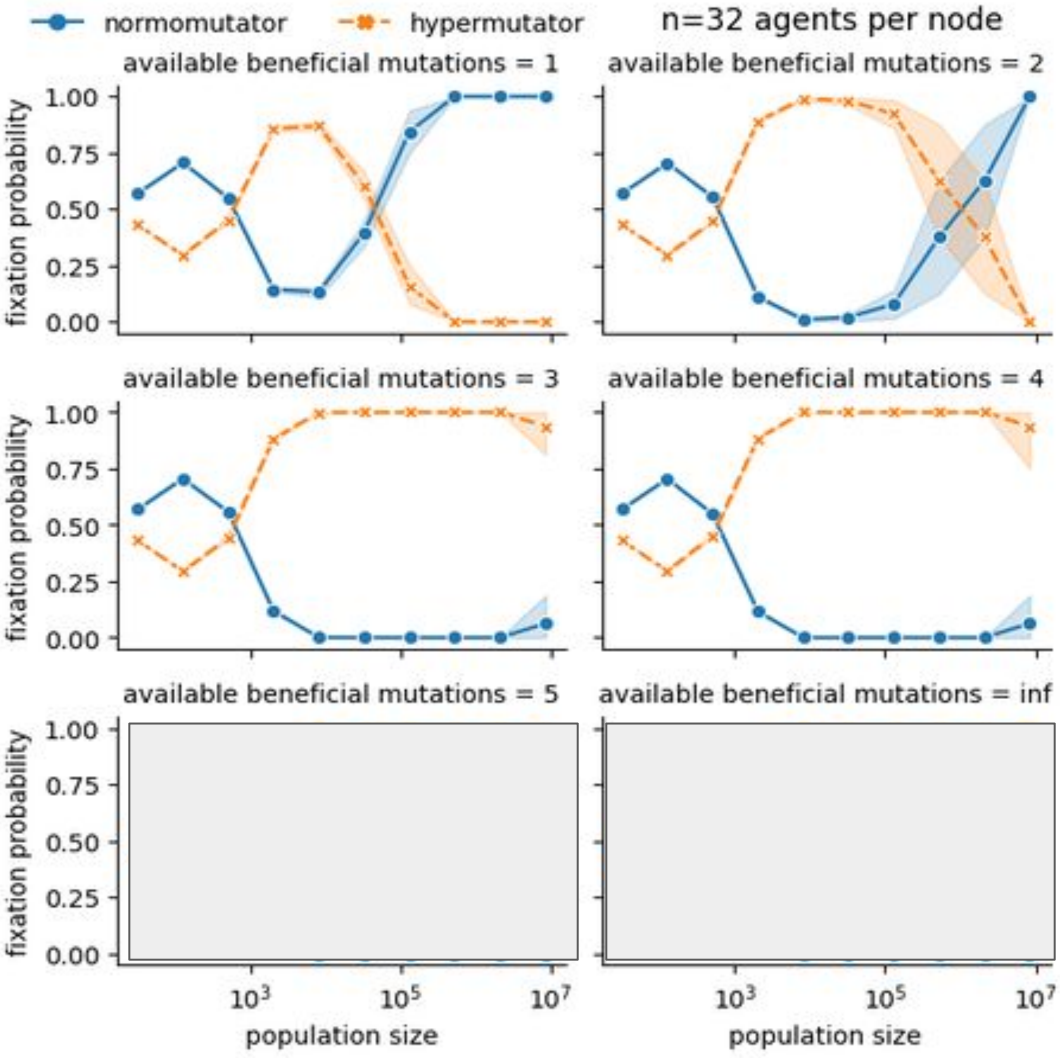

Hello!
STOP
0002
0002
queraltsciart


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

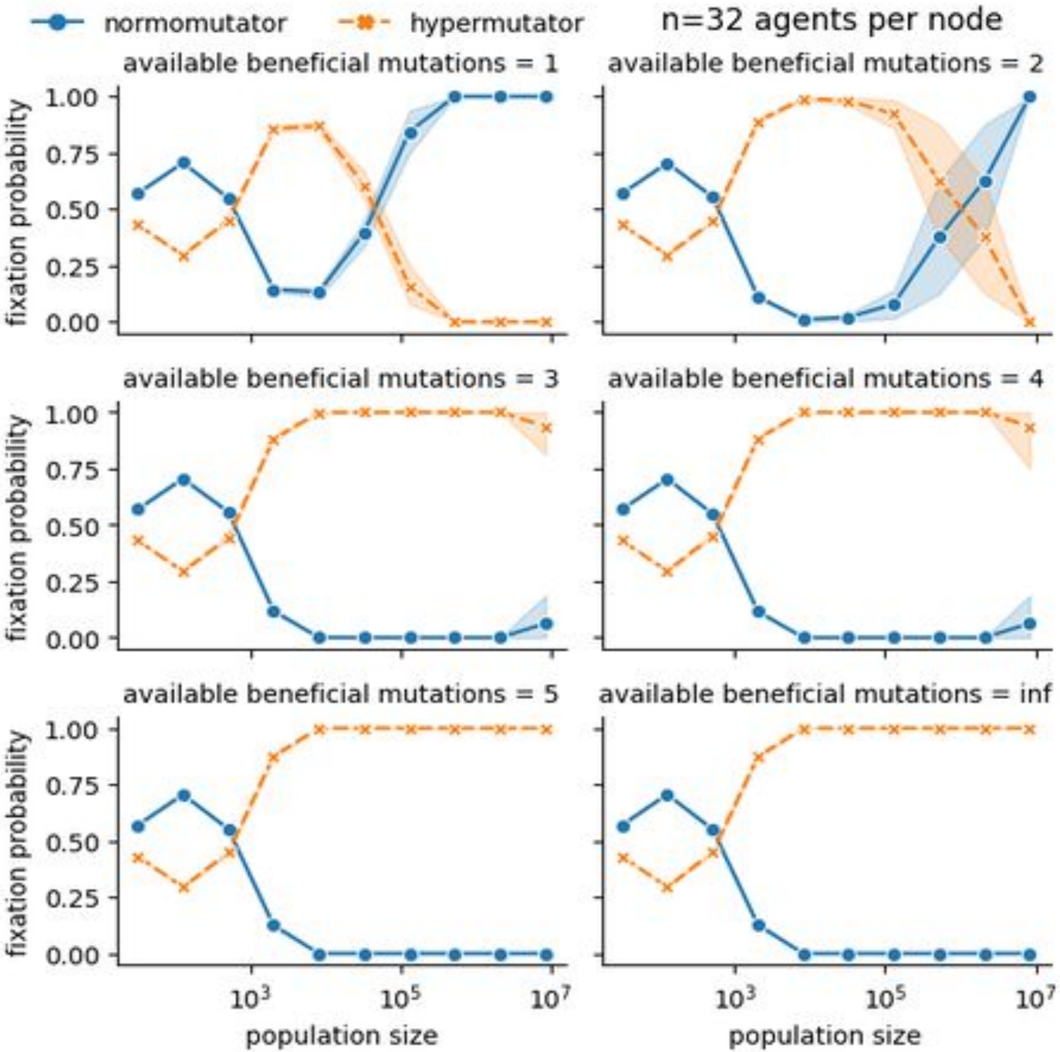

Hello!
STOP
0002
0002
queraltsciart


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

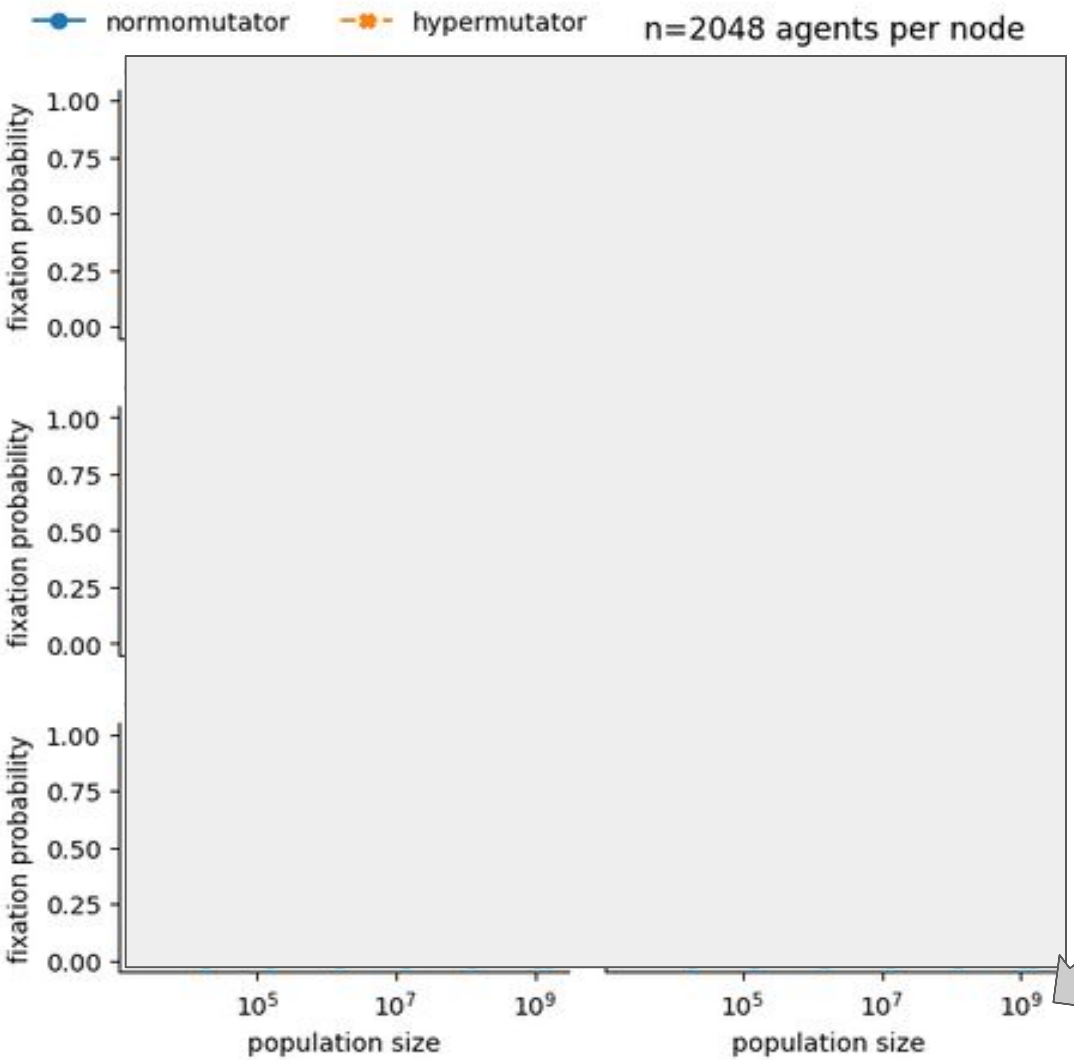

Hello!
STOP
0002
0002
queraltsciart


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

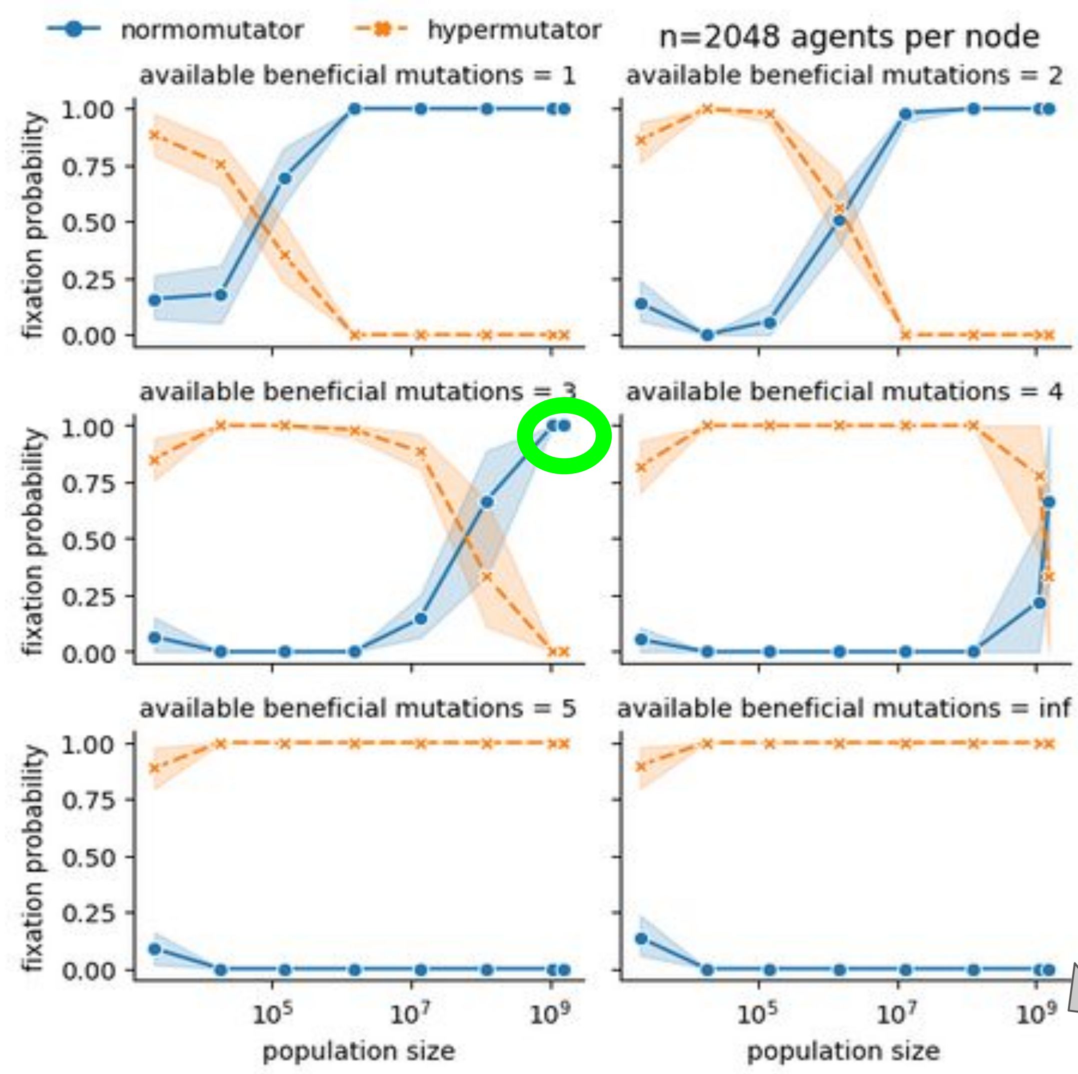

Hello!
STOP
0002
0002
queraltsciart


normomutator
hypermutator
n=32 agents per node
available beneficial mutations = 1
available beneficial mutations = 2
available beneficial mutations = 3
available beneficial mutations = 4
available beneficial mutations = 5
available beneficial mutations = inf
fixation probability
1.00
0.75
0.50
0.25
0.00
10^3
10^5
10^7
population size

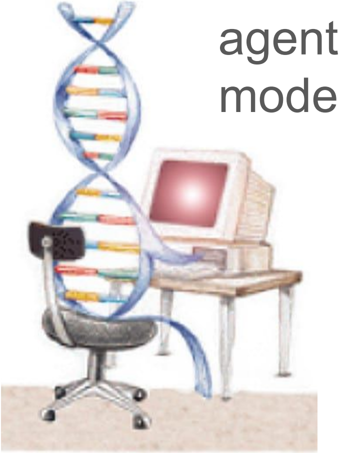

Hello!
queraltsciart
STOP
0002
0002



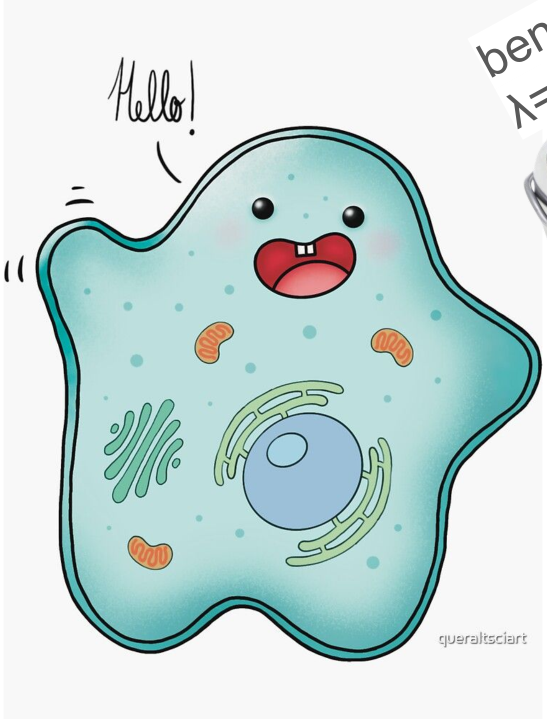
1 billion agents



1 billion agents



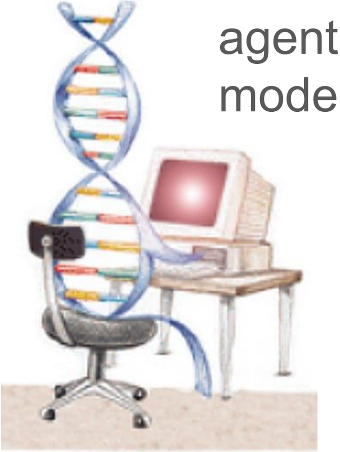
agent-based
model



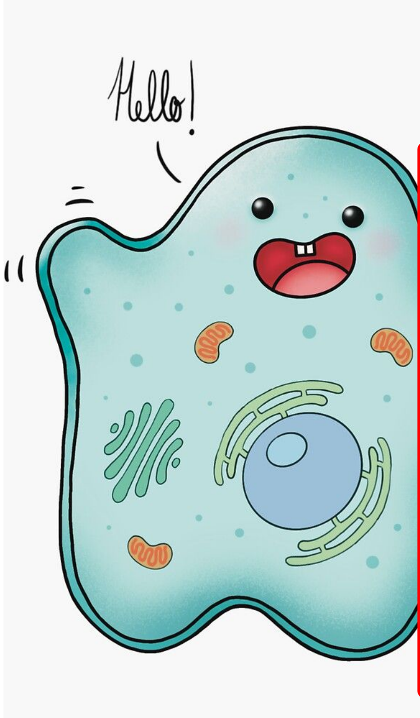
beneficial
λ=1 in 100k

deleterious
λ=1 in 1k

poisson distribution



agent-based
model

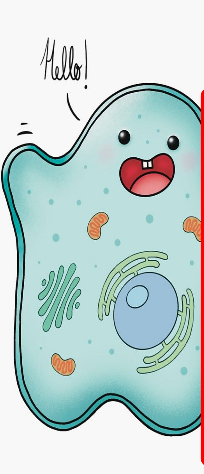


beneficial
$p=1$ in 100k

deleterious
$\lambda=1$ in 1k

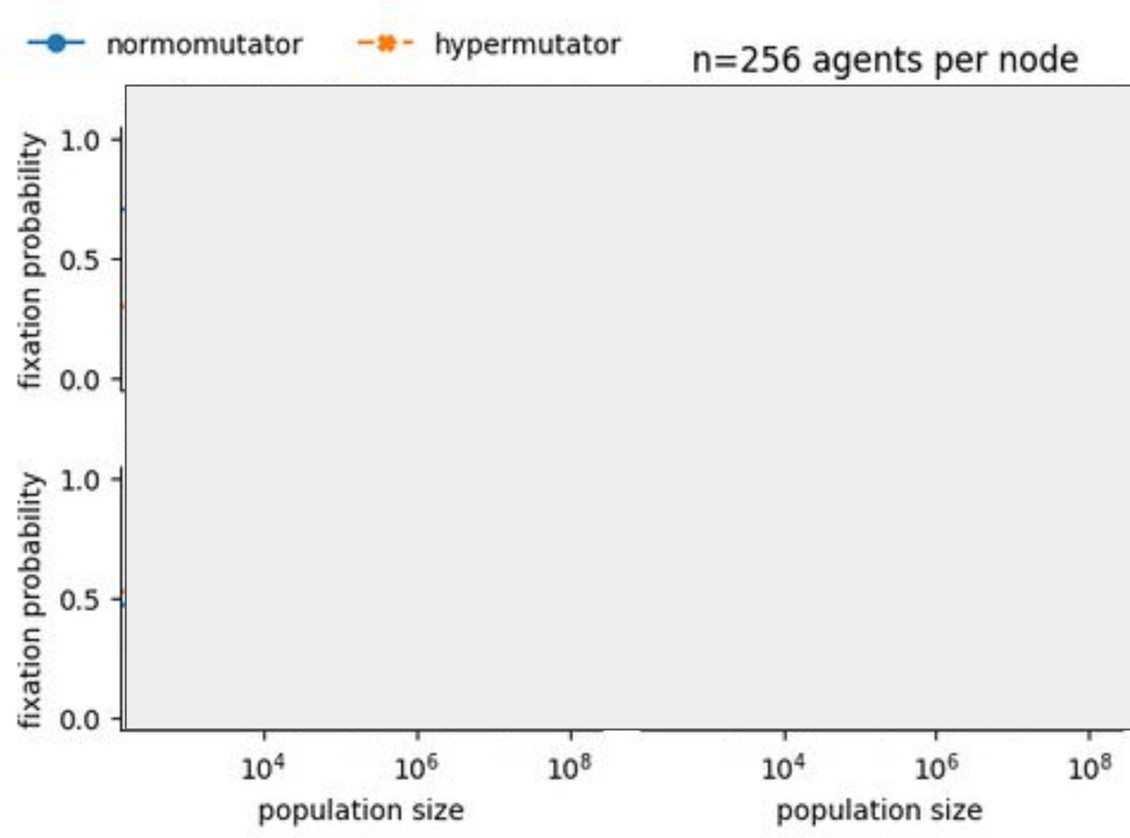
n **loci** available

n loci

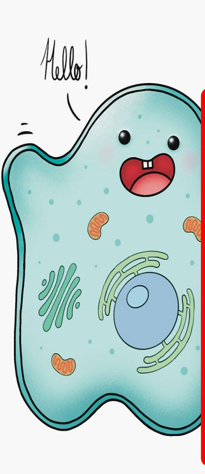

Hello!


United States of America Coin Collector's Map


0002

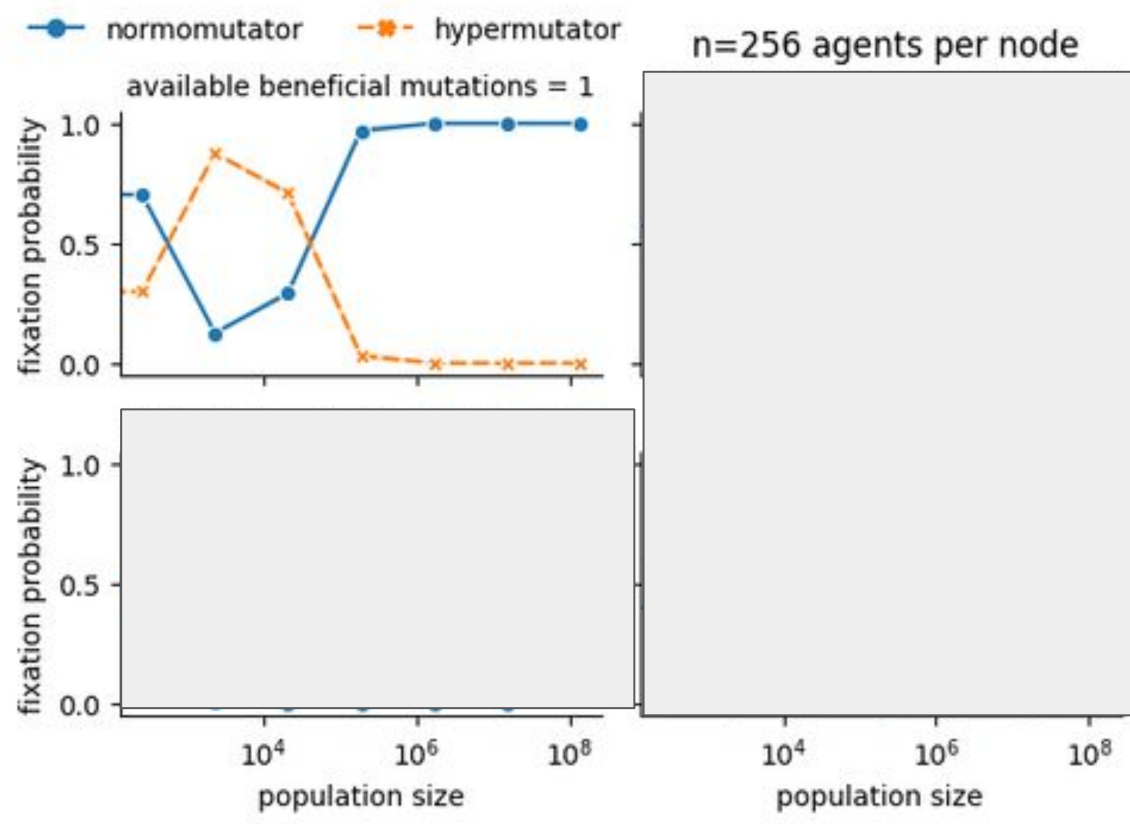

normomutator
hypermutator
n=256 agents per node
fixation probability
1.0
0.5
0.0
fixation probability
1.0
0.5
0.0
10^4
10^6
10^8
population size
10^4
10^6
10^8
population size

n loci

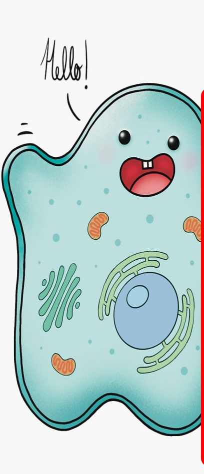

Hello!

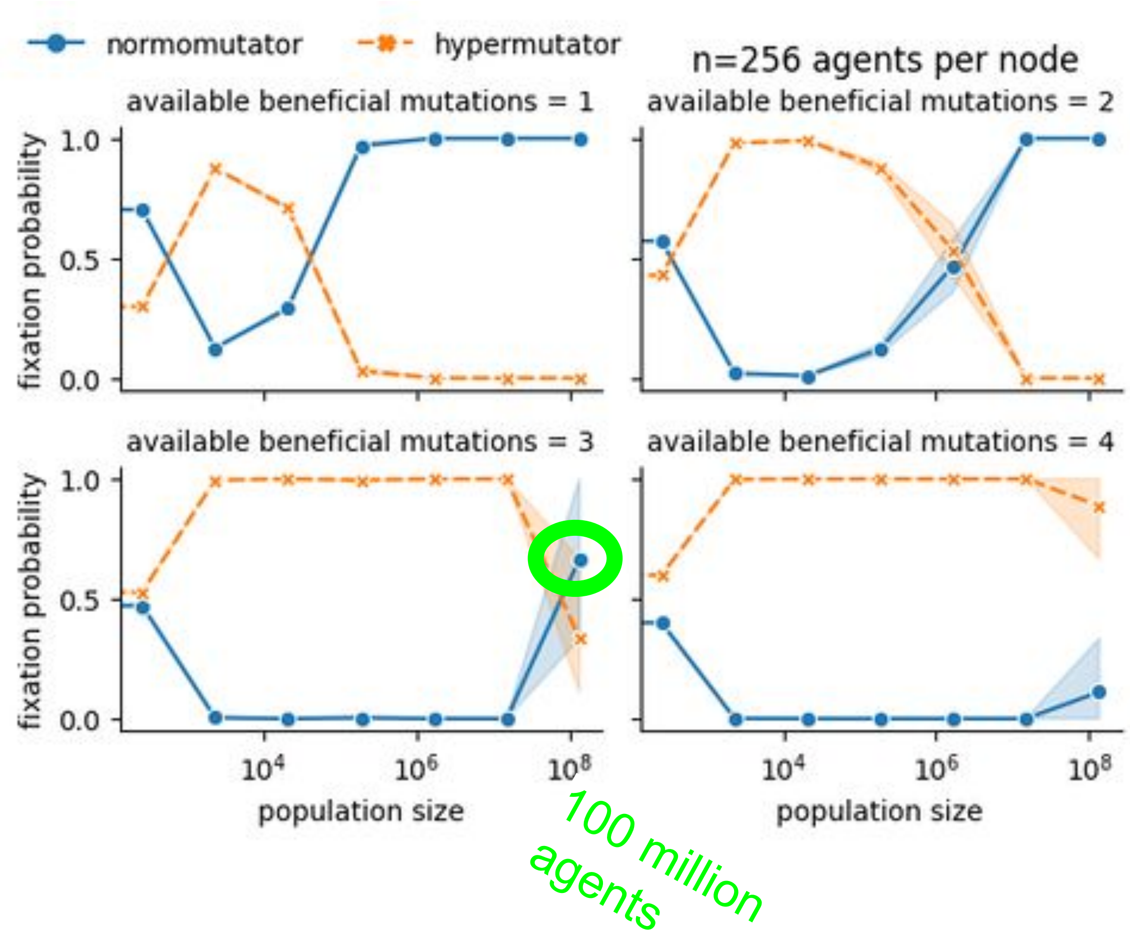

United States of America Coin Collector's Map


0002


normomutator
hypermutator
n=256 agents per node
available beneficial mutations = 1
fixation probability
1.0
0.5
0.0
fixation probability
1.0
0.5
0.0
10^4
10^6
10^8
population size
10^4
10^6
10^8
population size

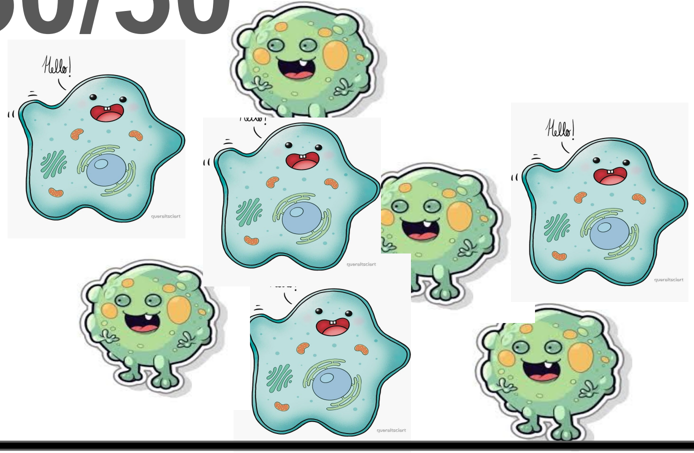
n loci

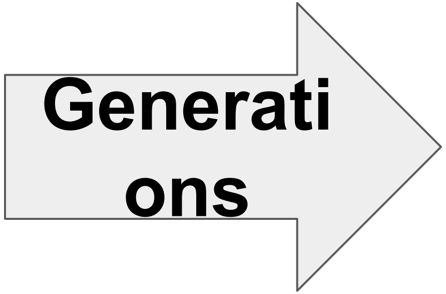




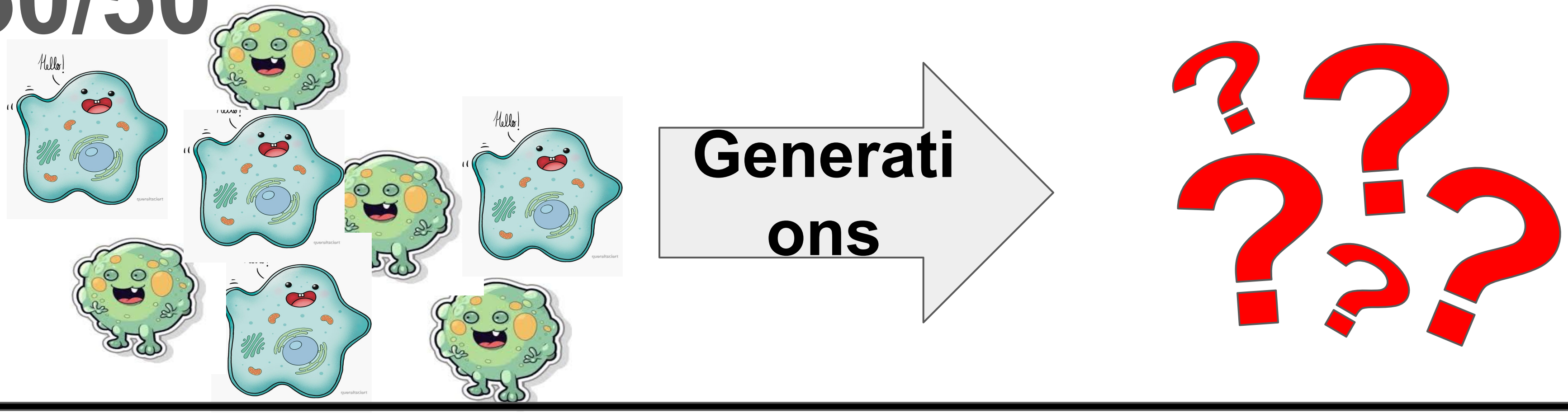
100 million agents

50/50




Generations

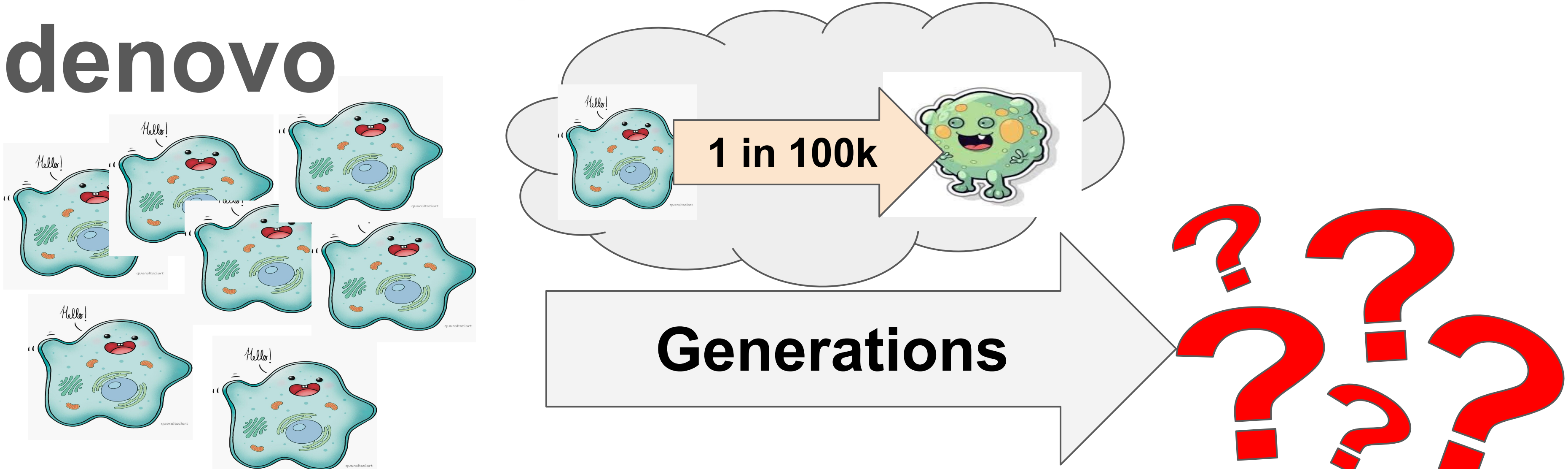
50/50



Generations

---

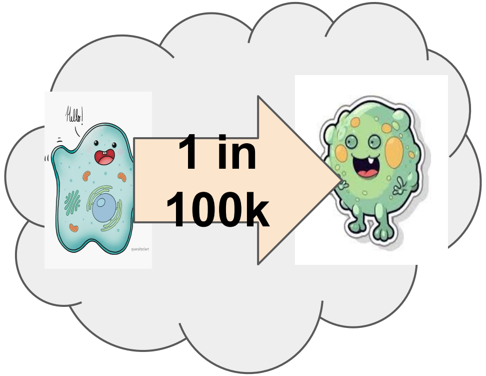
denovo

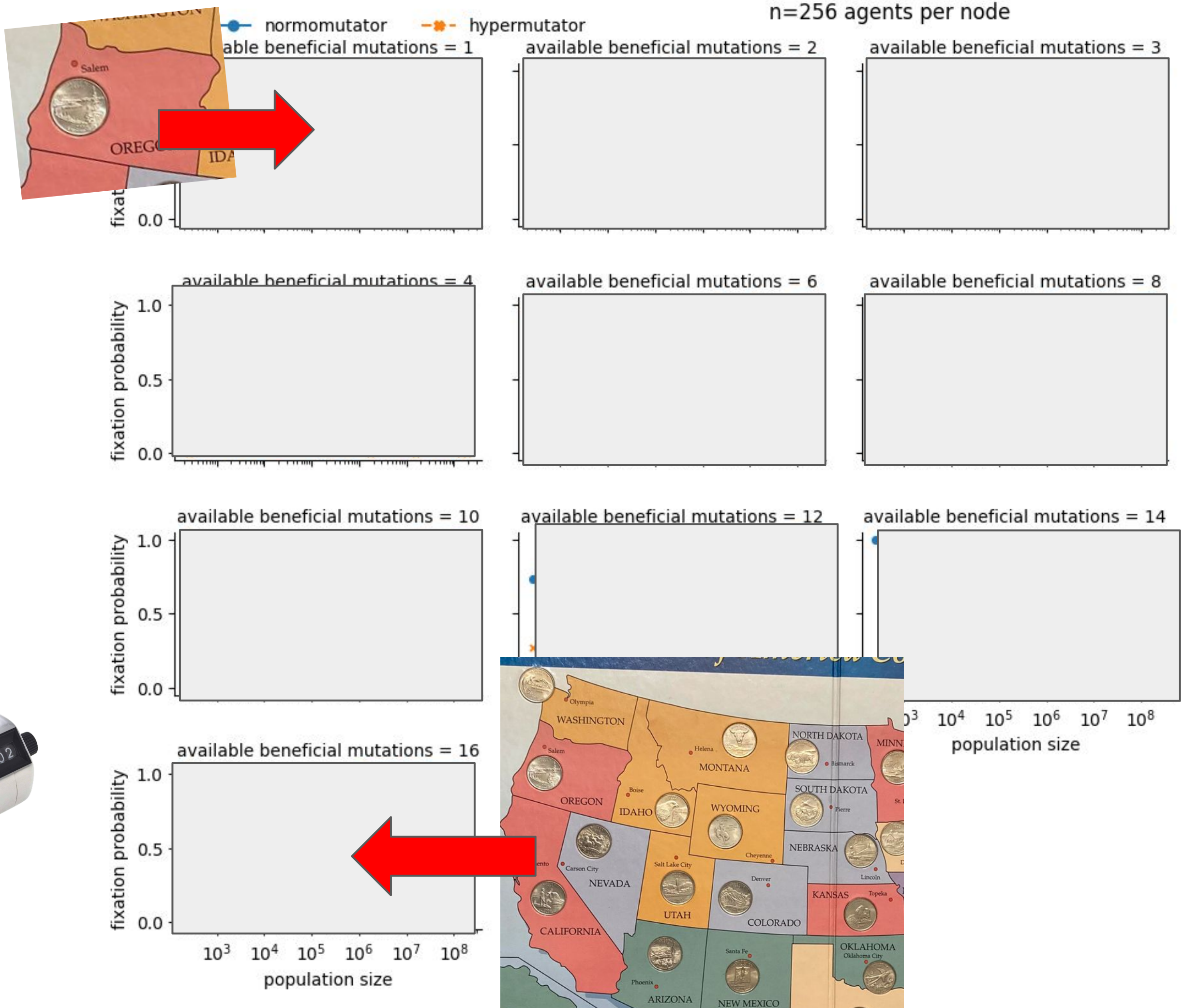


1 in 100k

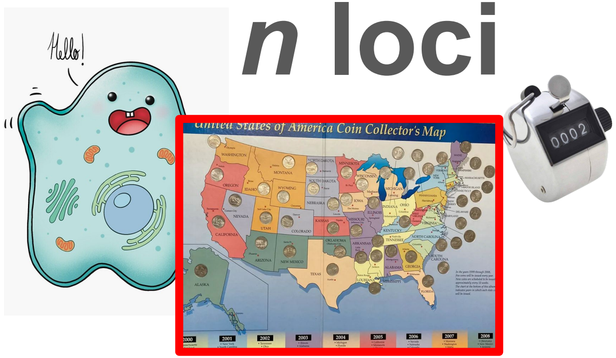
Generations

denovo

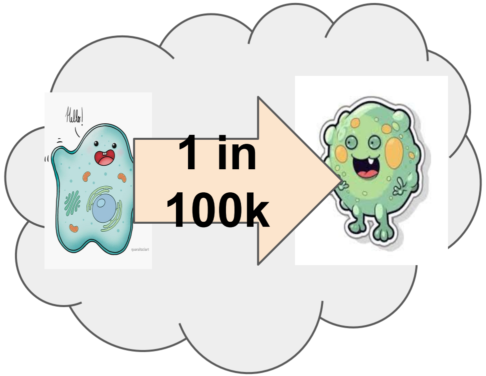


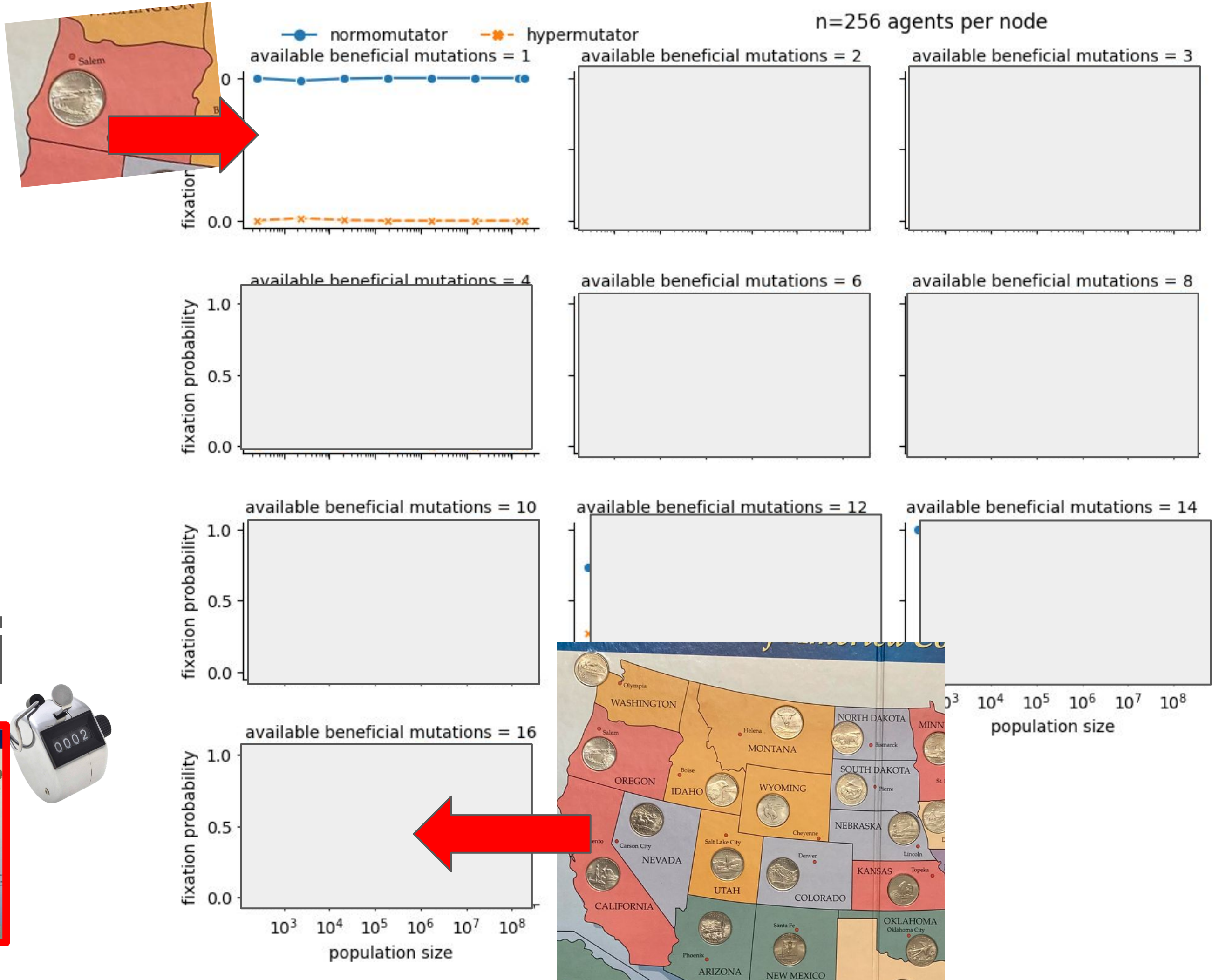


n loci

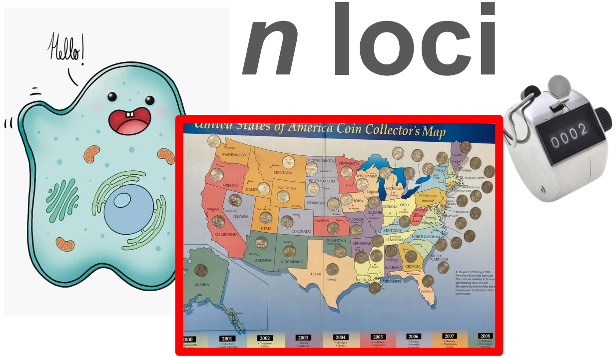


denovo

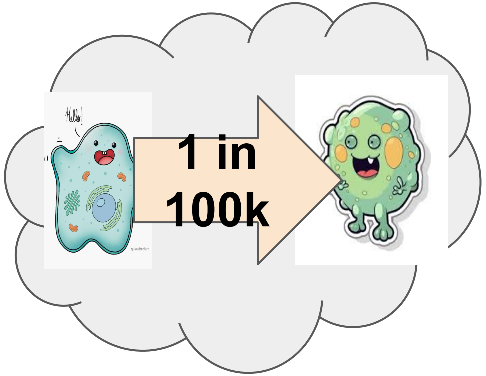


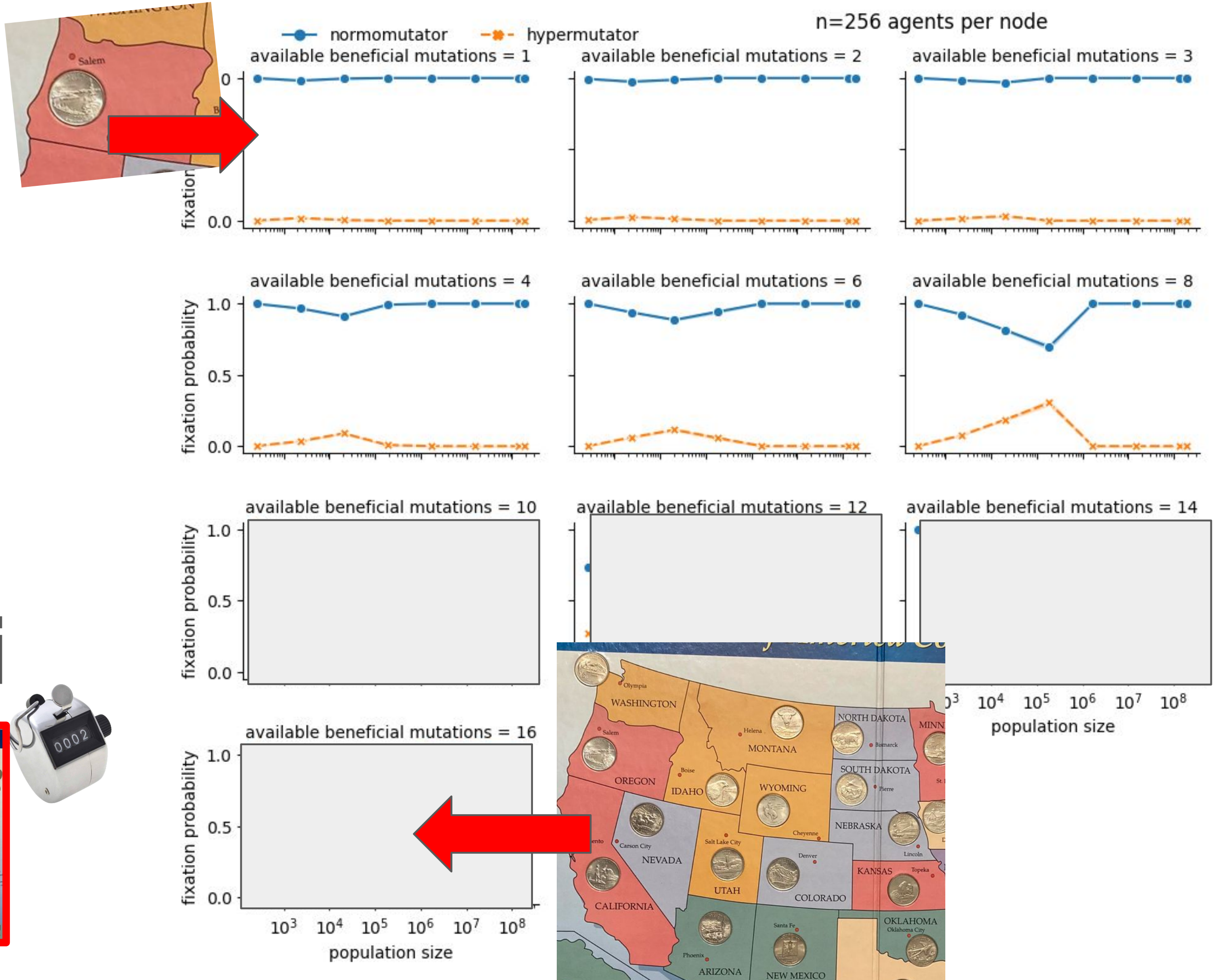


n loci

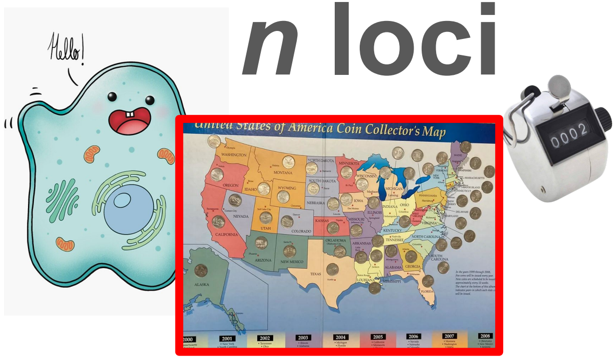


denovo

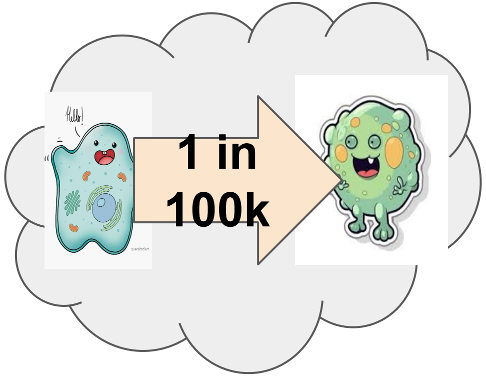


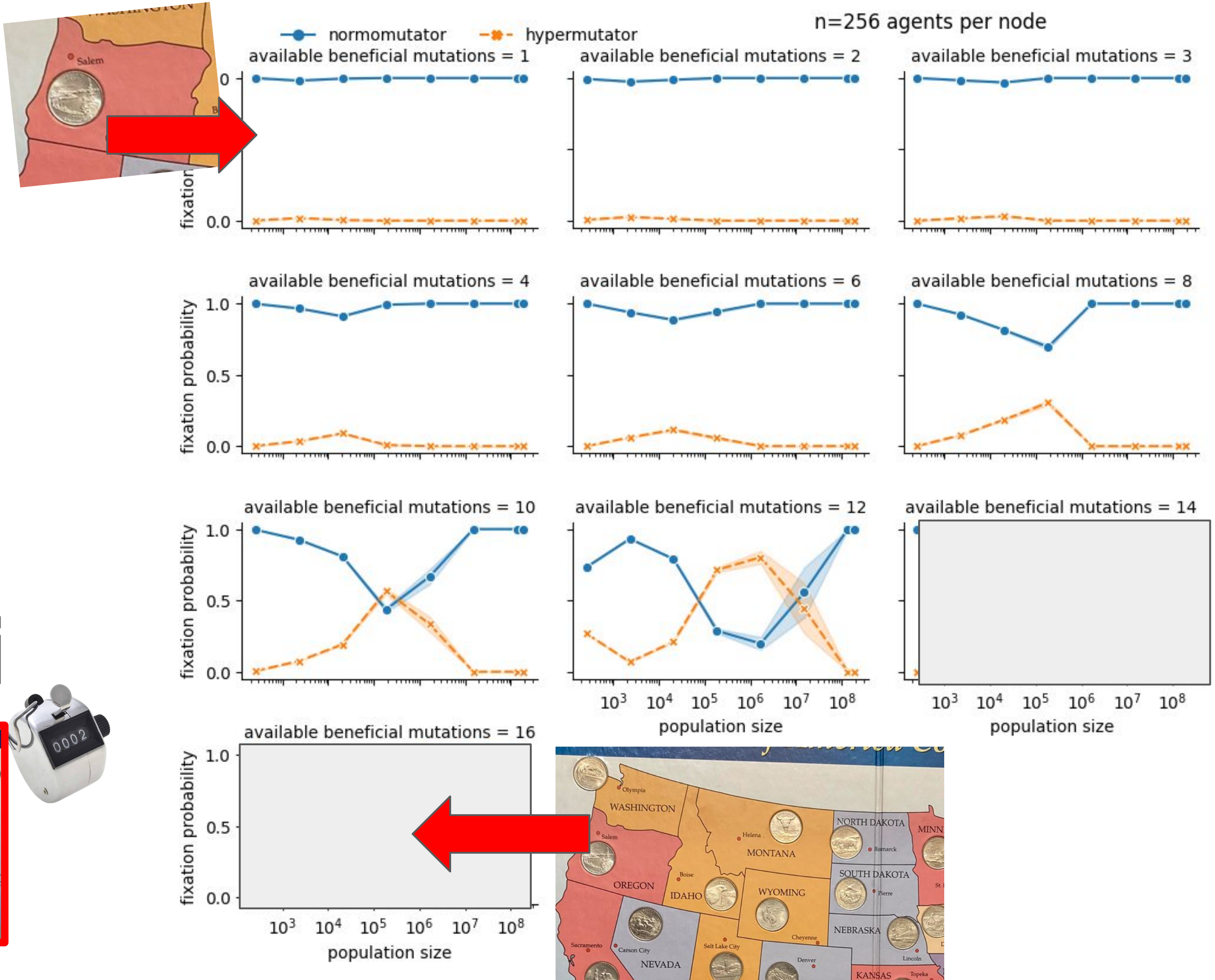


n loci

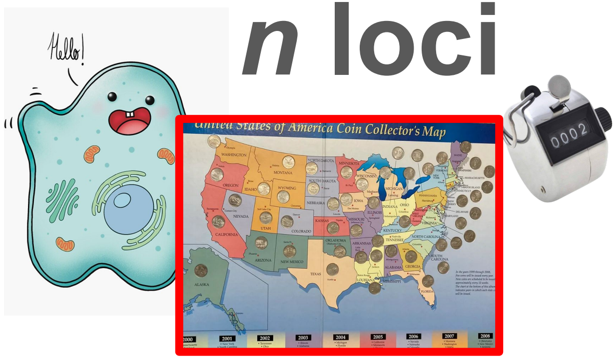


denovo

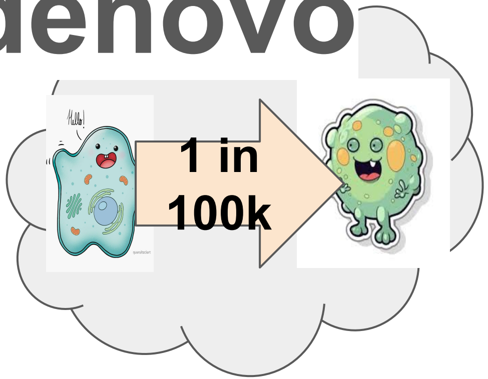


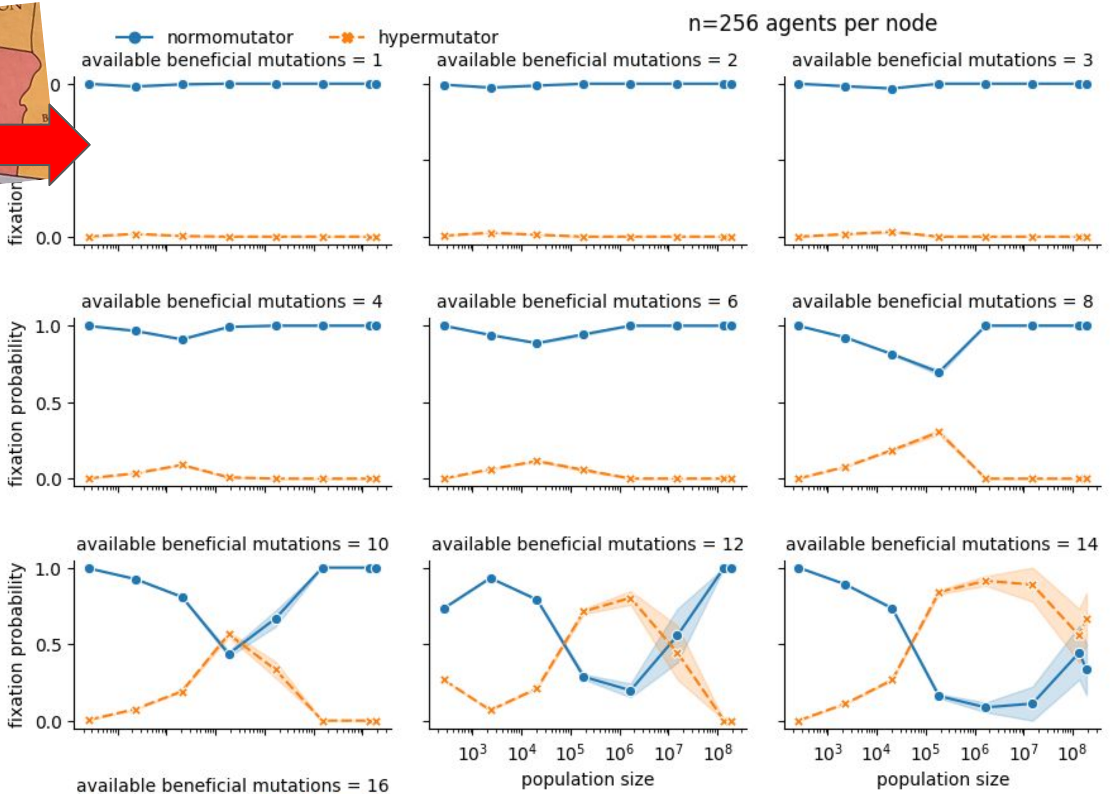


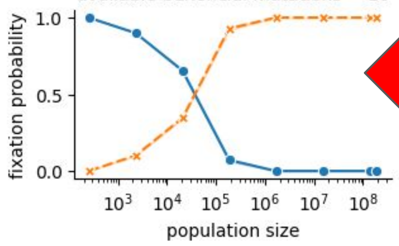
n loci



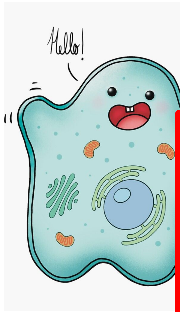
denovo



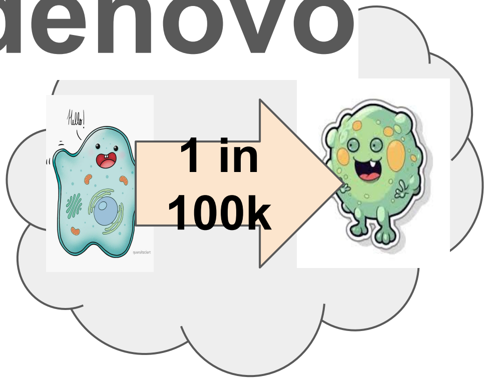


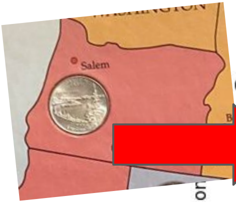


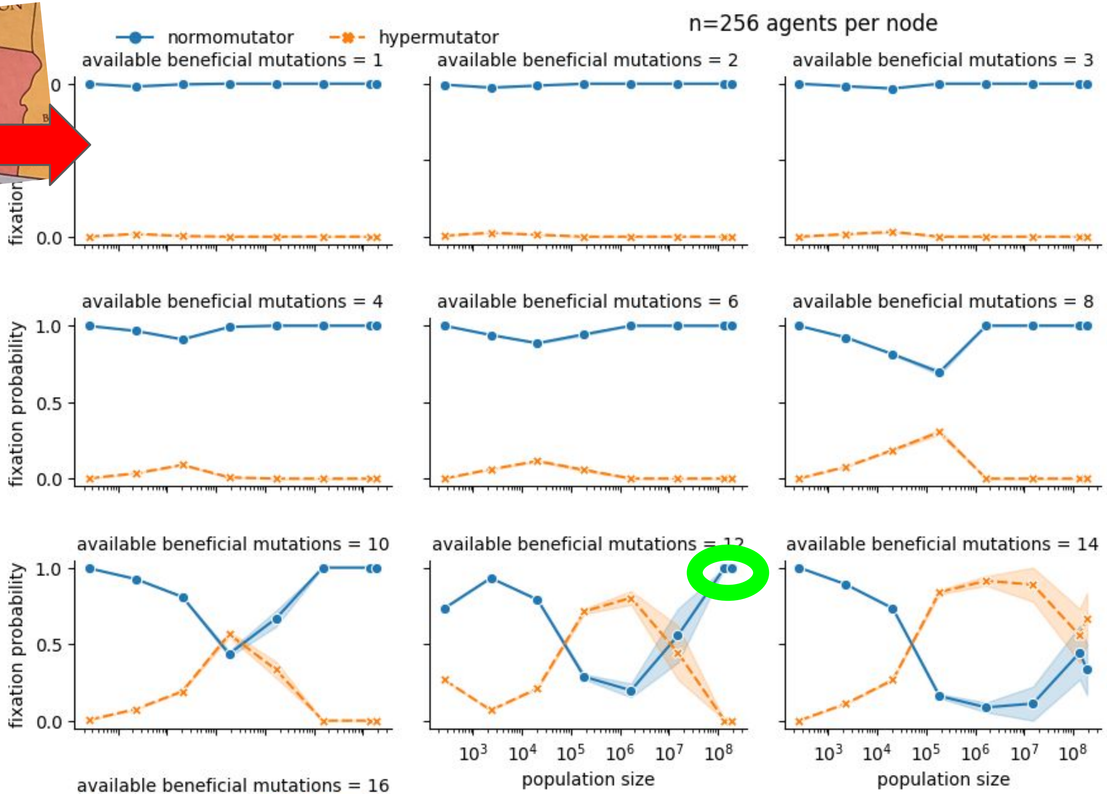
n loci

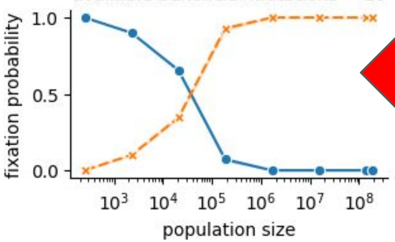


denovo

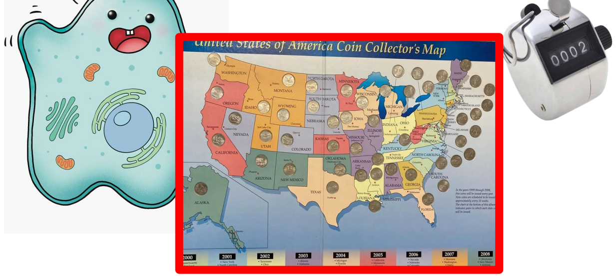




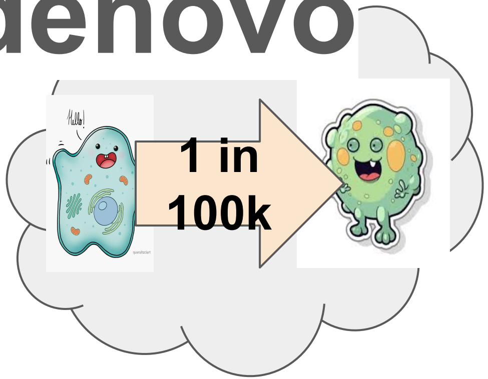




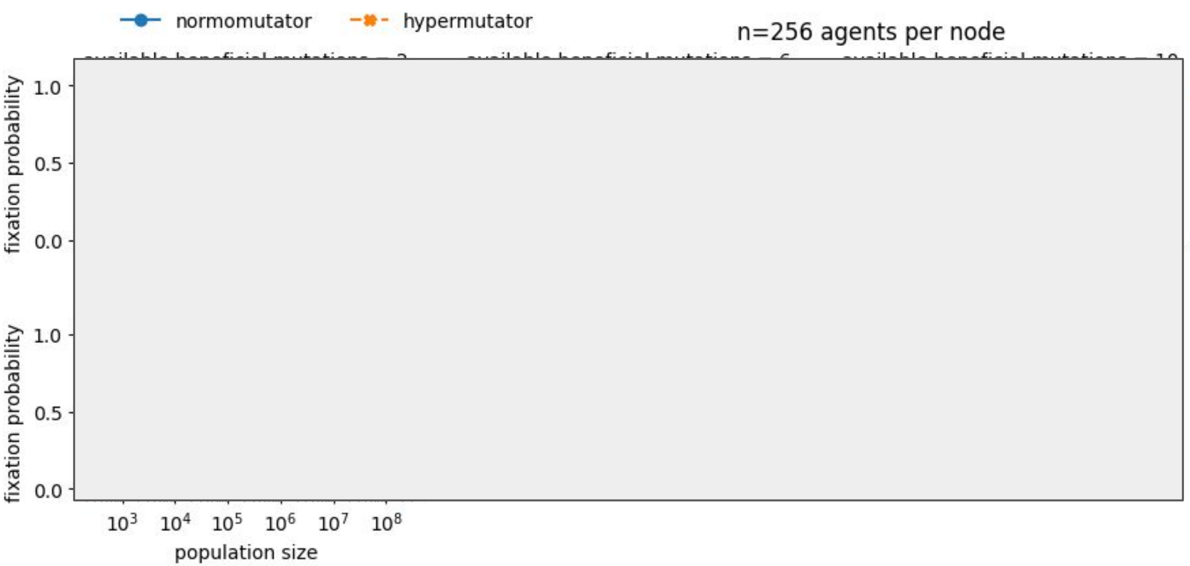
n loci



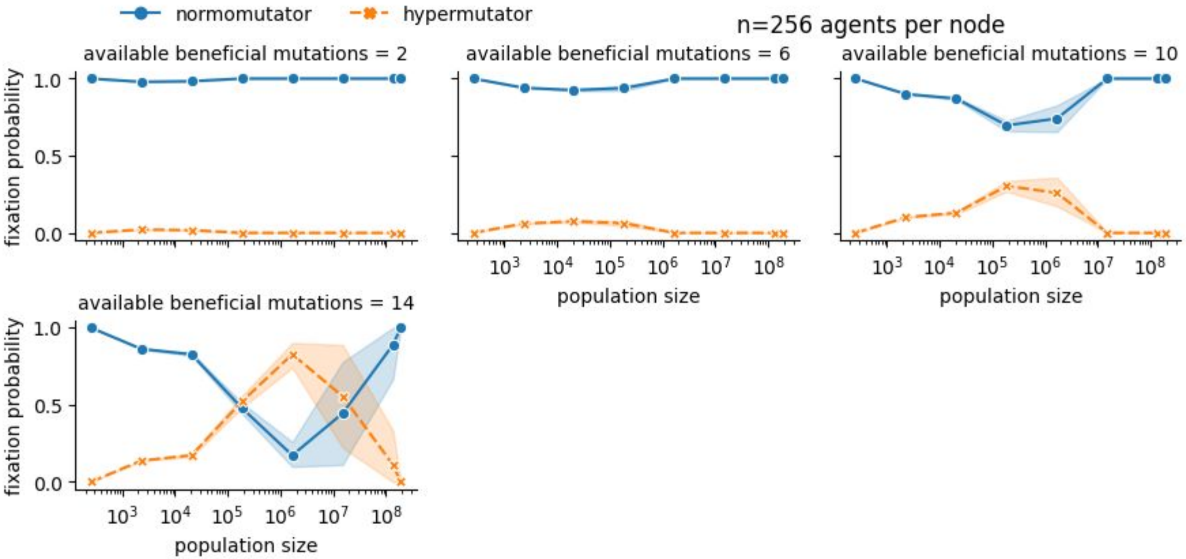
denovo


1 in 100k


Hello!
STOP
0002
0002


normomutator
hypermutator
n=256 agents per node
fixation probability
1.0
0.5
0.0
fixation probability
1.0
0.5
0.0
10^3
10^4
10^5
10^6
10^7
10^8
population size

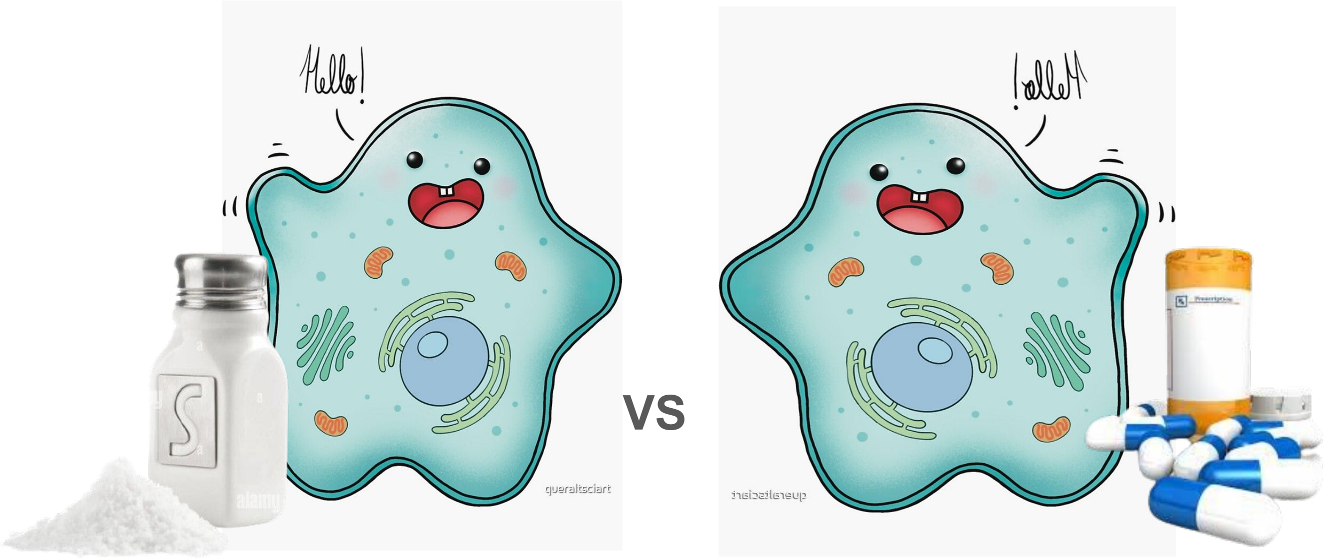
n=256 agents per node



Next Steps

Next Steps

Callens et al., 2023

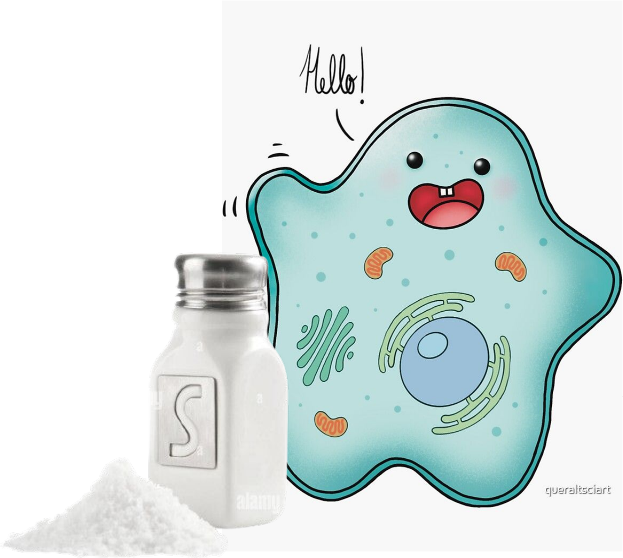


osmotic pressure

antibiotics

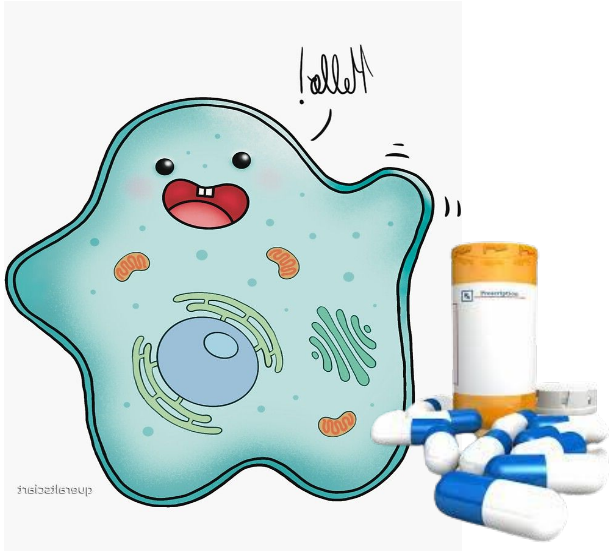
Next Steps

Callens et al., 2023



osmotic pressure
more adaptive loci
more hypermutators

VS



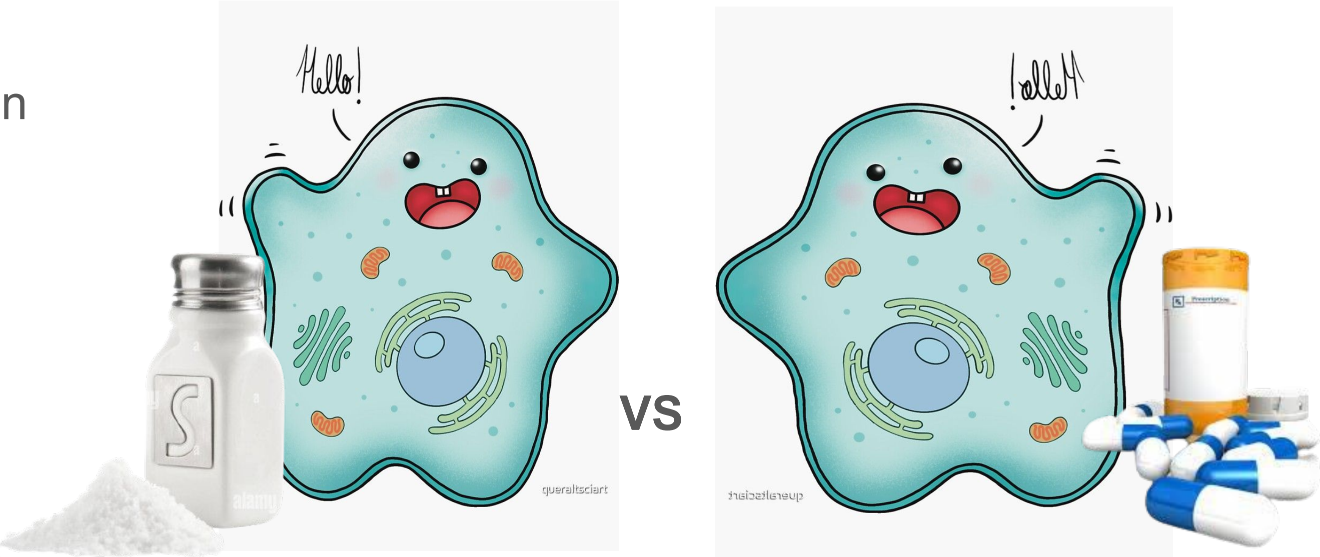
antibiotics
fewer adaptive loci
fewer hypermutators

Q: how many beneficial muts per strain?

Next Steps

- more sophisticated fitness landscape models
- well-mixed ABM on GPU
- analyze time to fixation in model vs. theory expectation
- create mathematical model for *de novo* selection curves
- connect to data from Callens et al or LTEE?

Callens et al., 2023

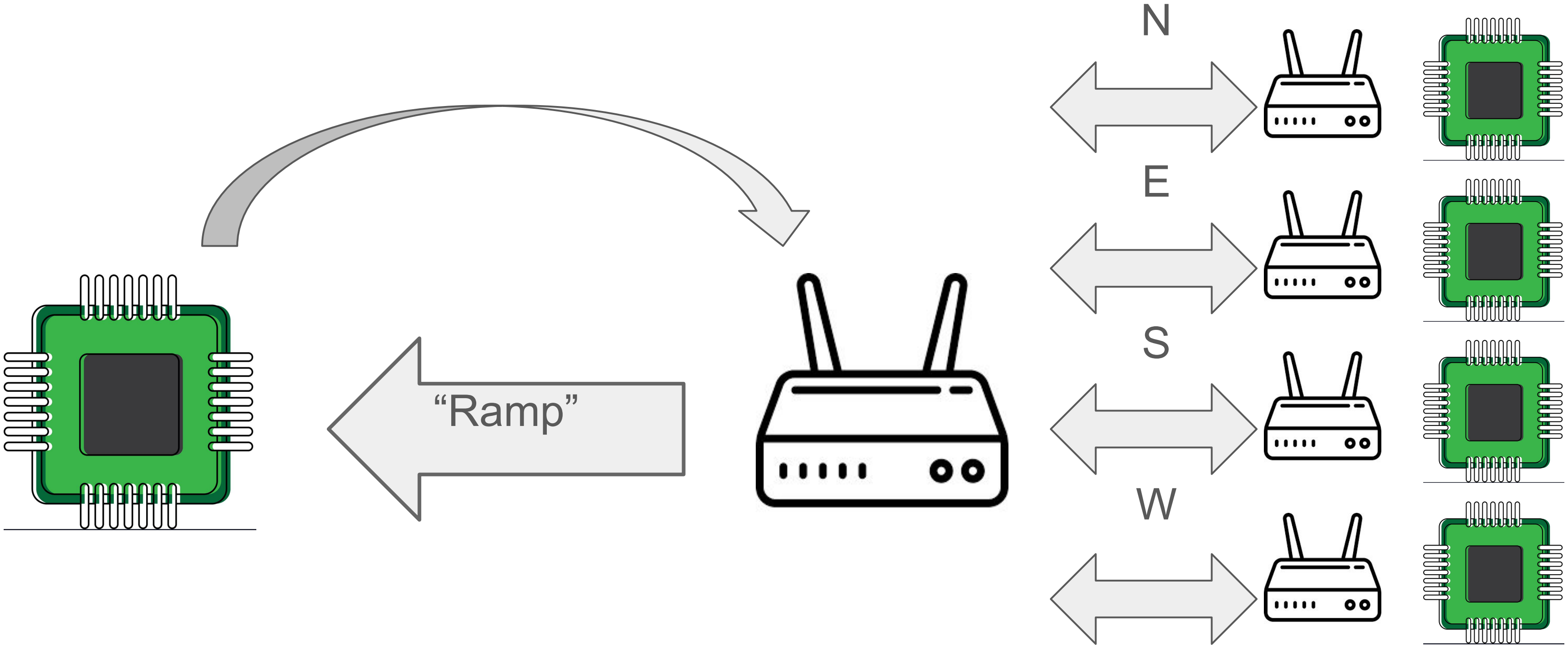


osmotic pressure
more adaptive loci
more hypermutators

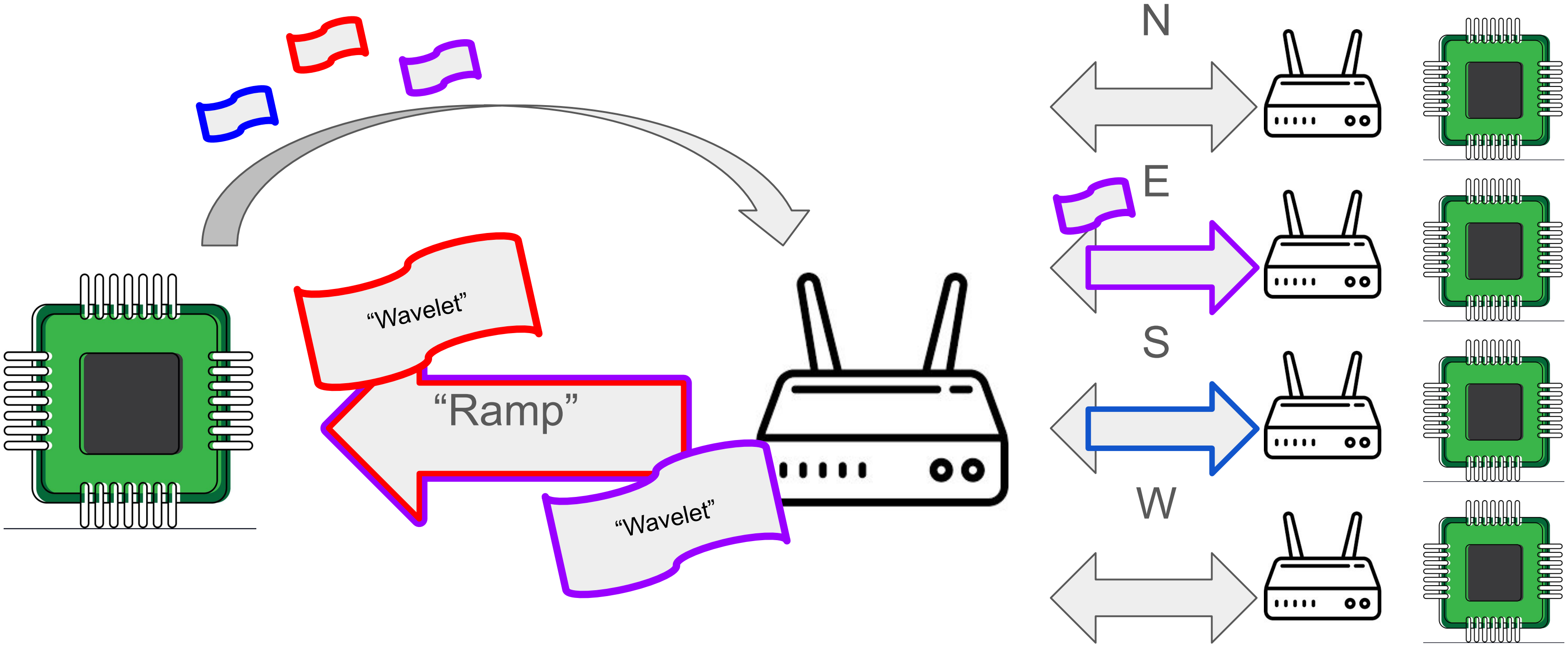
antibiotics
fewer adaptive loci
fewer hypermutators

Q: how many beneficial muts per strain?

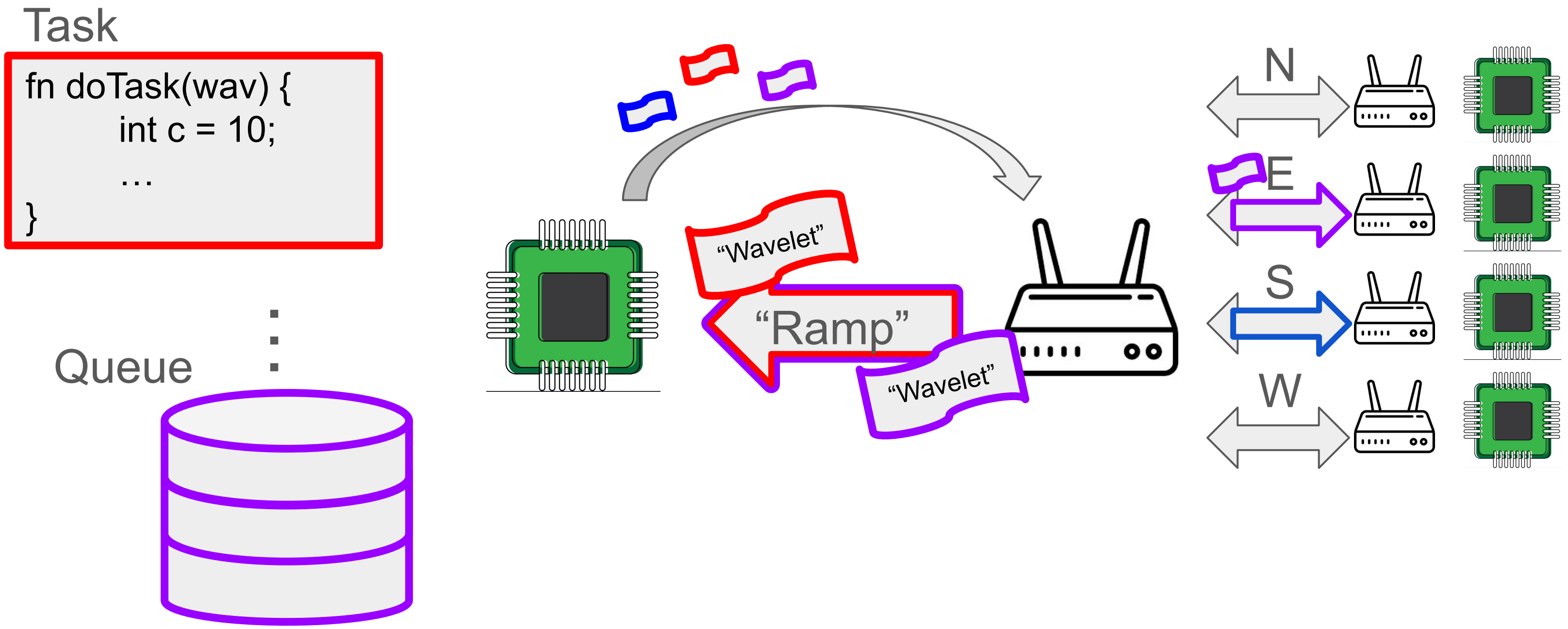
Tagged, Event-driven Programming Model



Tagged, Event-driven Programming Model



Tagged, Event-driven Programming Model

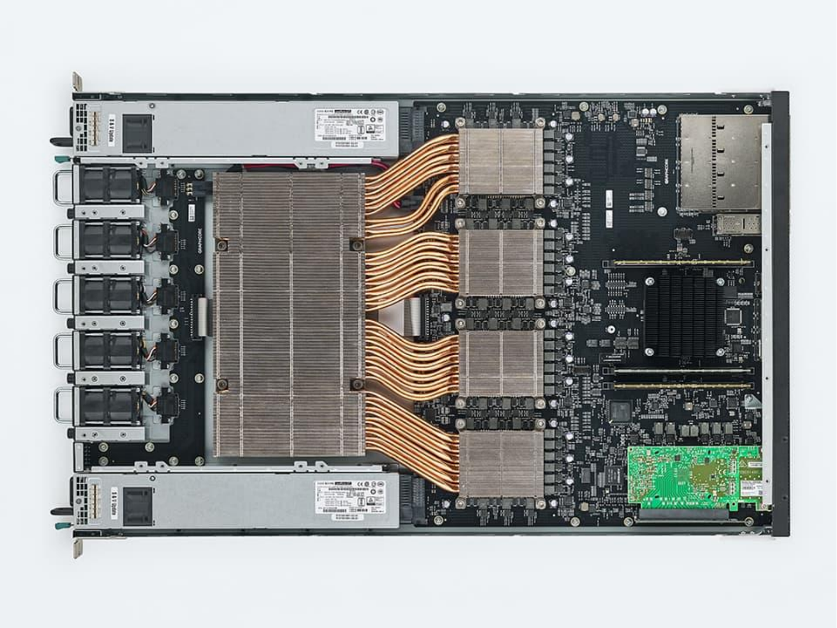


Task

```
fn doTask(wav) {
        int c = 10;
        ...
}
```

[more about WSE programming model in pocket slides]

GraphCore IPU — another AI/ML accelerator



- 1,200 cores per chip
- clustered up to 1,024 chips

morenoma@umich.edu

Part 1: On-hardware Experiment

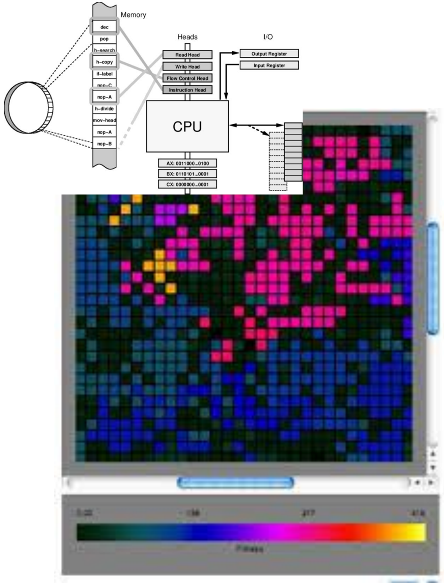
Scale and Digital Evolution

LTEE
(Good et al., 2017)

Image credit: Credit: National Institute of Allergy and Infectious Diseases, National Institutes of Health

Scale and Digital Evolution

LTEE
(Good et al., 2017)



Avida
(Ofria and Wilke, 2009)

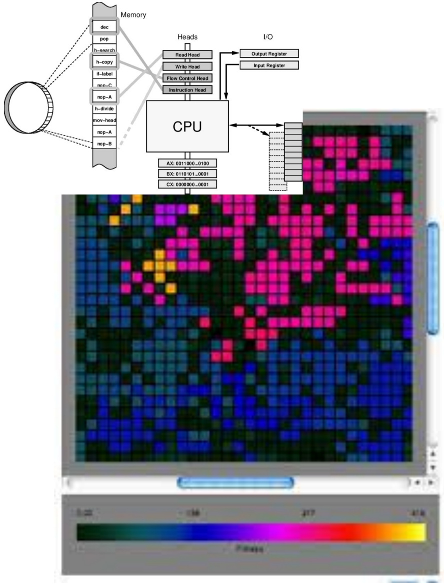
📚 [these slides] https://hopth.ru/ea

📧 morenoma@umich.edu

Image credit: Credit: National Institute of Allergy and Infectious Diseases, National Institutes of Health

Scale and Digital Evolution

LTEE
(Good et al., 2017)



Avida
(Ofria and Wilke, 2009)

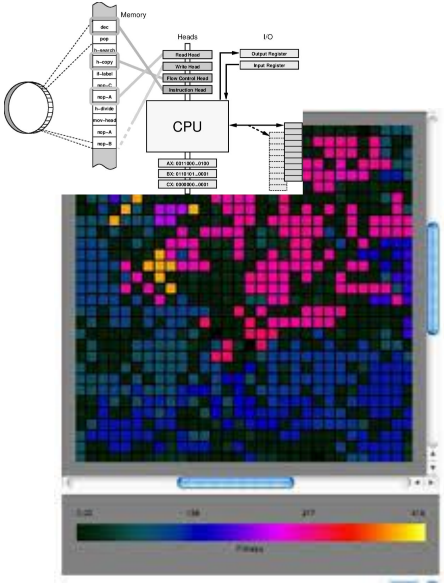
- both ~billion replications/day

📚 [these slides] https://hopth.ru/ea

📧 morenoma@umich.edu

Scale and Digital Evolution

LTEE
(Good et al., 2017)



Avida
(Ofria and Wilke, 2009)

- both ~billion replications/day

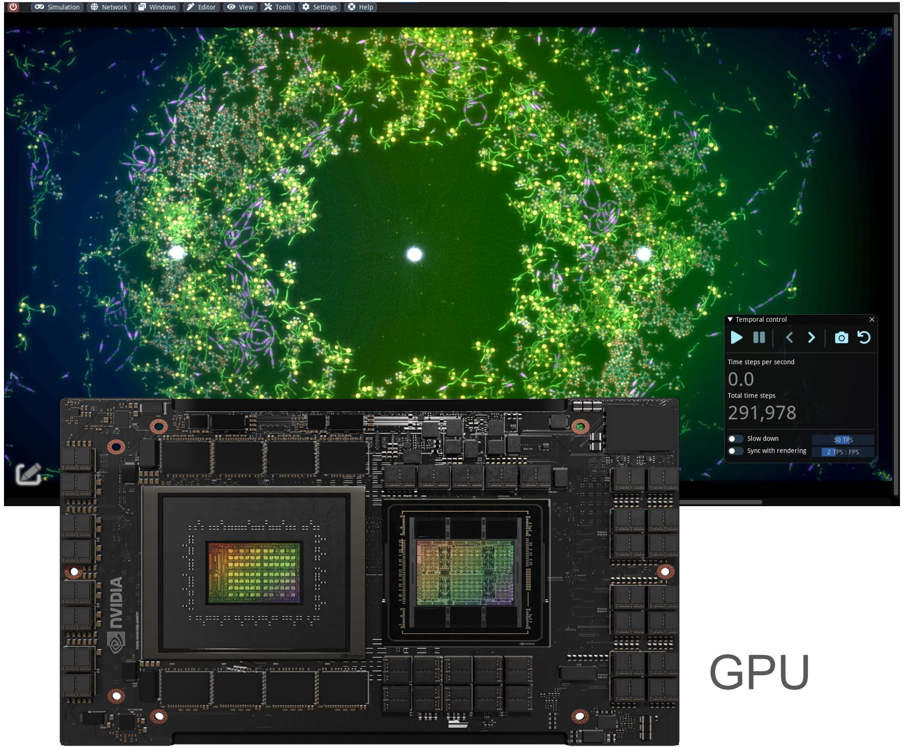
Cross-scale Phenomena:

- ecological communities
- multicellularity/major transitions

📚 [these slides] https://hopth.ru/ea

📧 morenoma@umich.edu

ALIEN Project
(Heinemann, 2024)

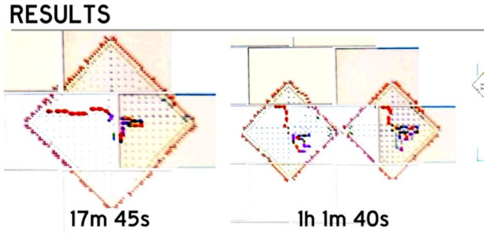


Time steps per second
0.0
Total time steps
291,978

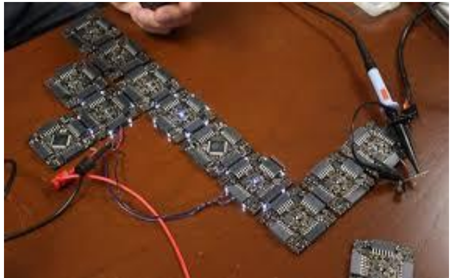
GPU

📚 [slides, paper, code] https://hopth.ru/de

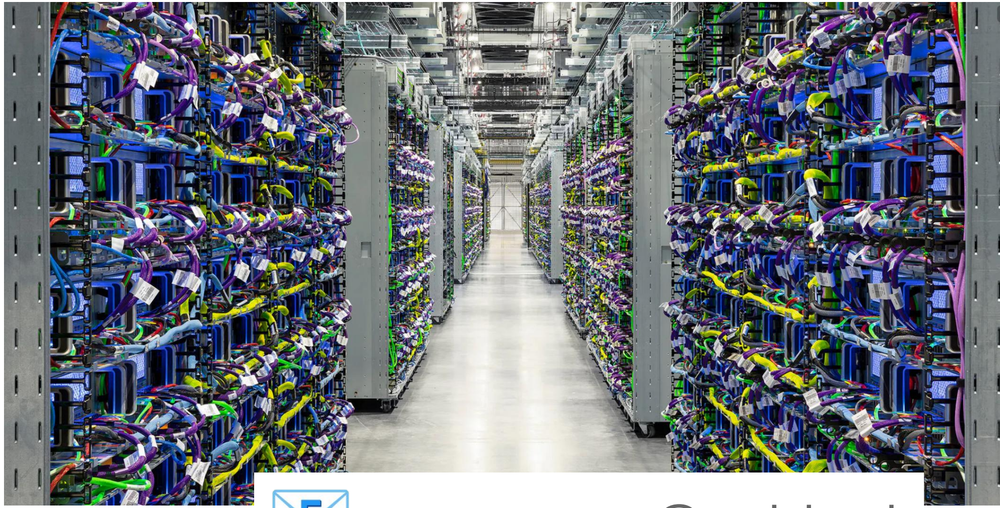
MFM/ULAM/T2 Tiles
(Ackley, 2023)



Illuninato x Machina
(Ackley, 2010)



Objective: Develop Methods to Harness Next Generation AI/ML Hardware for Agent-based Evolution Experiments



morenoma@umich.edu

Goal: develop
next-generatio
computing ha
agent-based e

Round-Up Part 2

Nordkyn X-back

Nordkyn Freighting

Nooksack Split Chest X-back

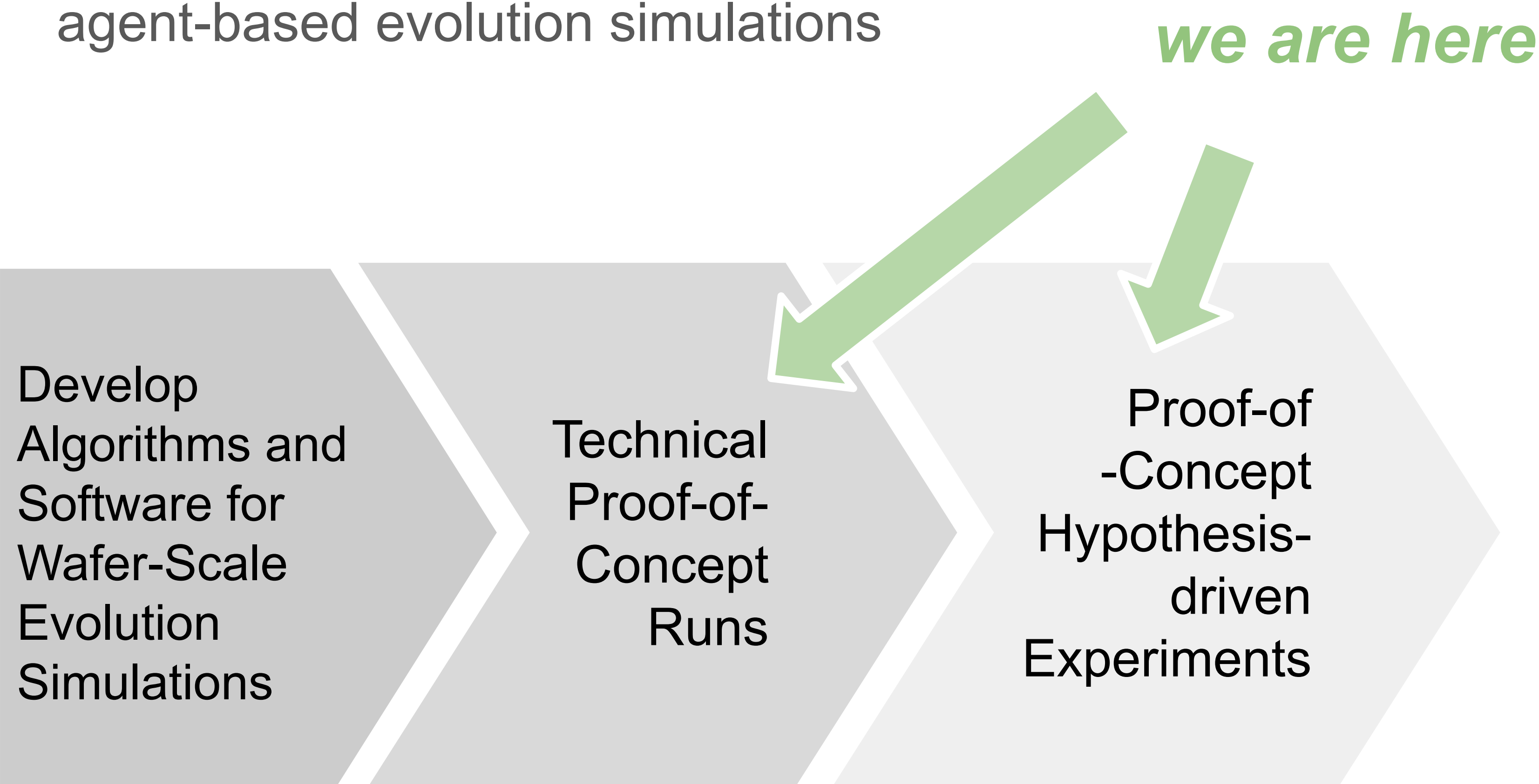
Nooksack Lightweight Standard X-back

Alpine Outfitters Sprint Racing X-back

Neesha X-back Featherweight

Neesha Adjustable Freight

Goal: develop methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations



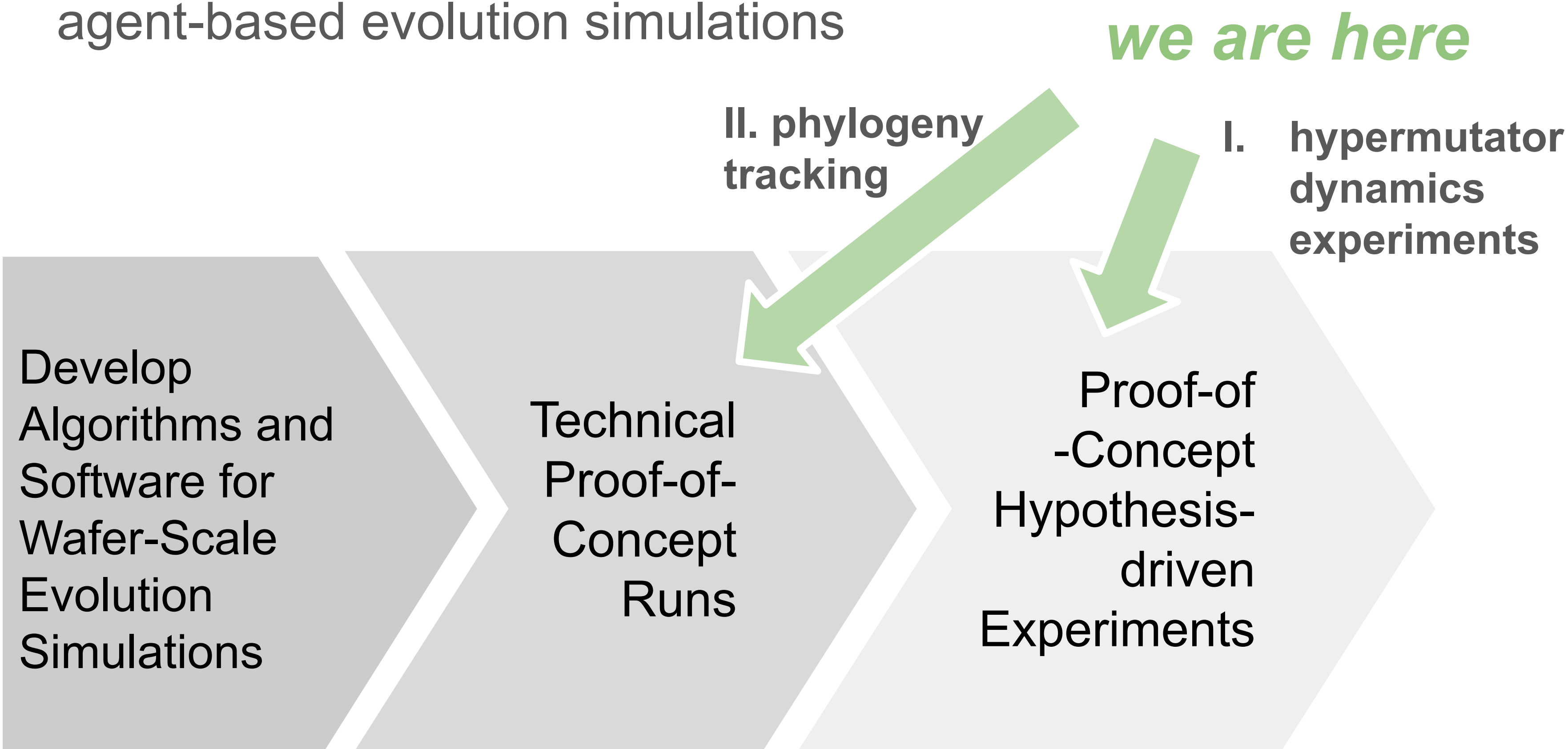
we are here

Develop Algorithms and Software for Wafer-Scale Evolution Simulations

Technical Proof-of-Concept Runs

Proof-of-Concept Hypothesis-driven Experiments

Goal: develop methods to harness next-generation high-performance computing hardware to enable larger agent-based evolution simulations



we are here

II. phylogeny tracking

I. hypermutator dynamics experiments

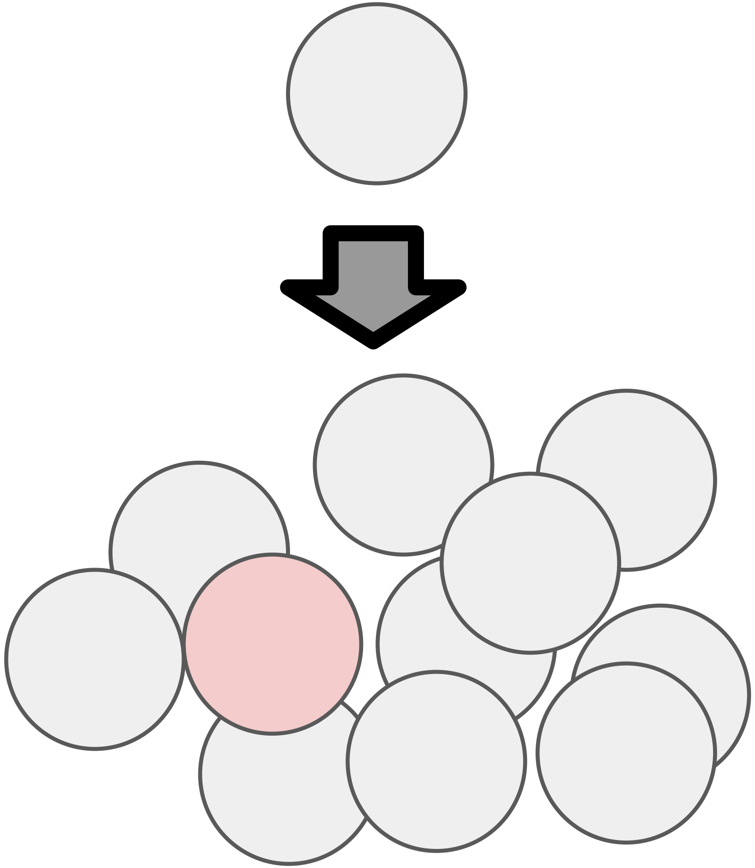
Develop Algorithms and Software for Wafer-Scale Evolution Simulations

Technical Proof-of-Concept Runs

Proof-of-Concept Hypothesis-driven Experiments

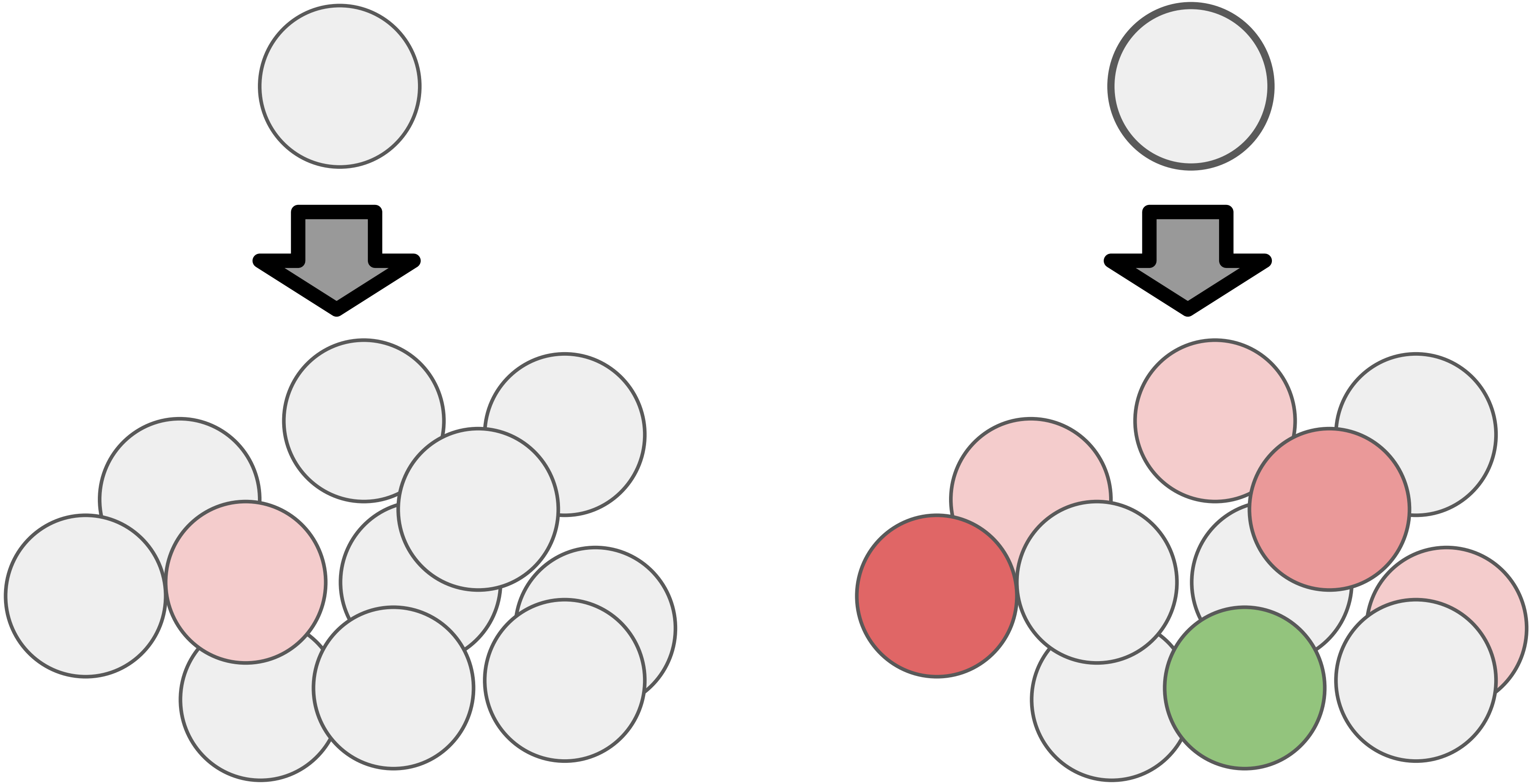
Part 1:

Hypermutator Dynamics Experiments on the Wafer-scale Engine

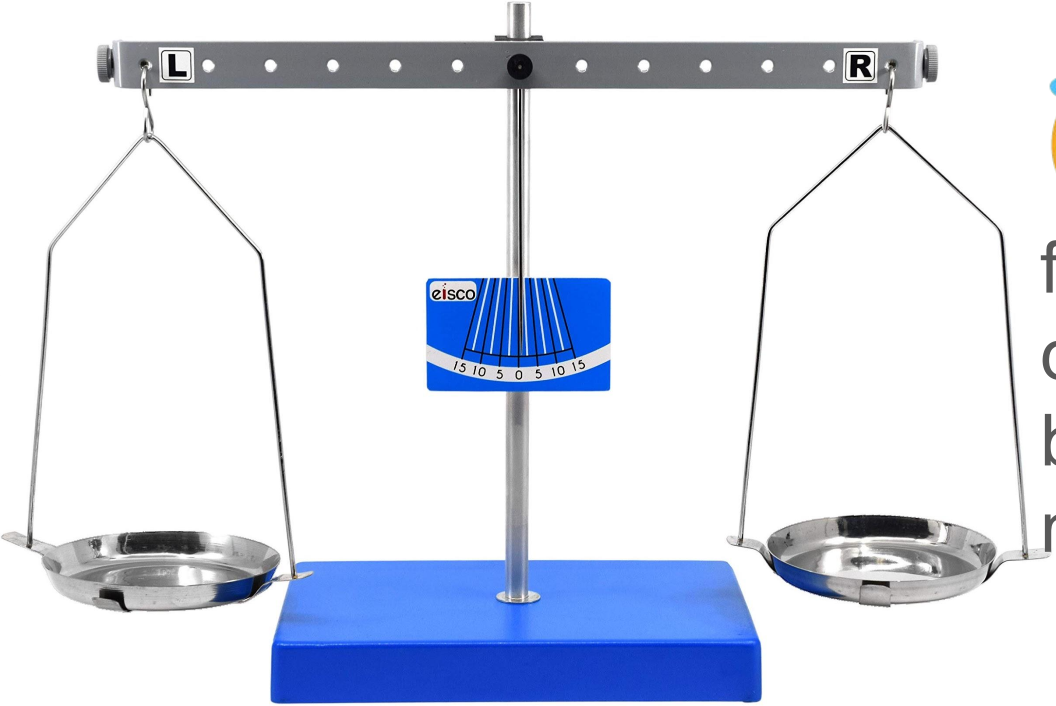


morenoma@umich.edu

“normomutator” **“hypermutator”**



hypermutator trait



😈 −

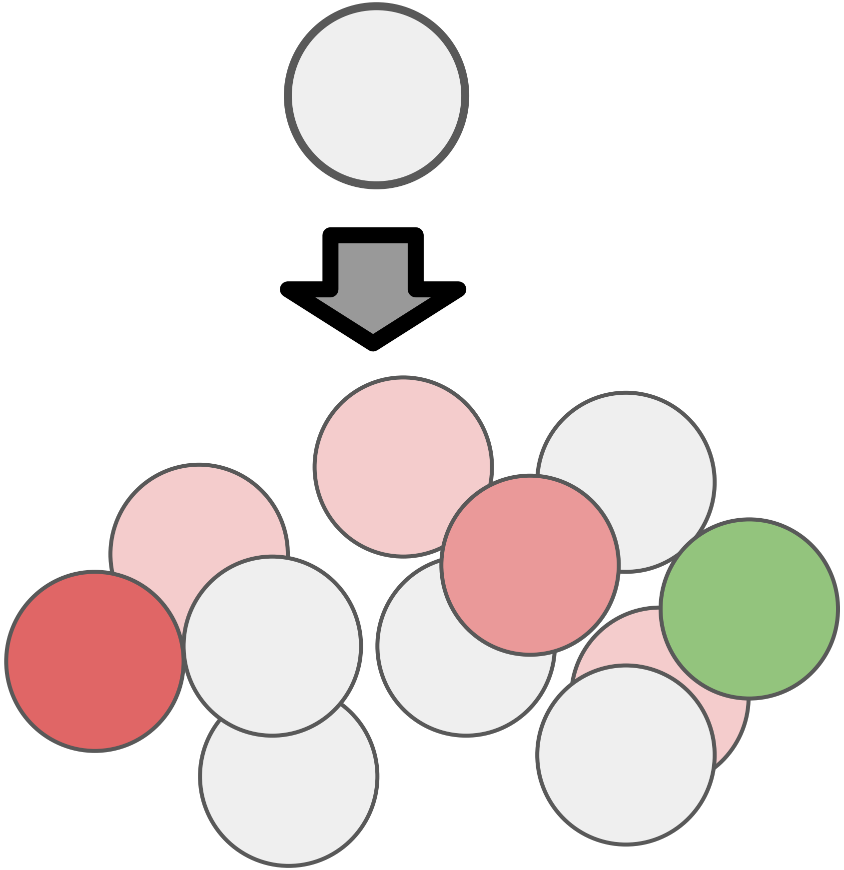
higher deleterious mutation load

😇 +

faster discovery beneficial mutations

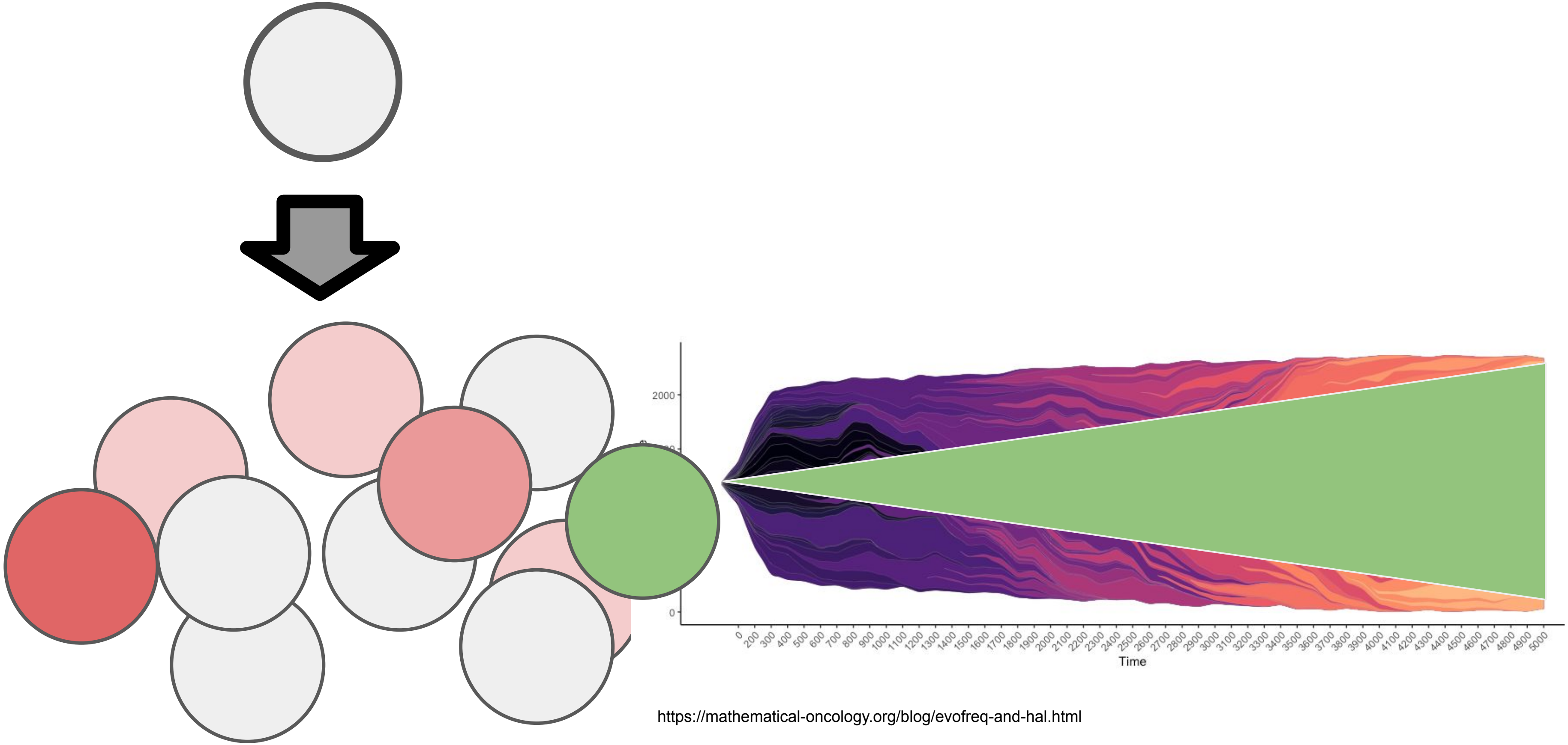
https://mathematical-oncology.org/blog/evofreq-and-hal.html

“hypermutator”



https://mathematical-oncology.org/blog/evofreq-and-hal.html

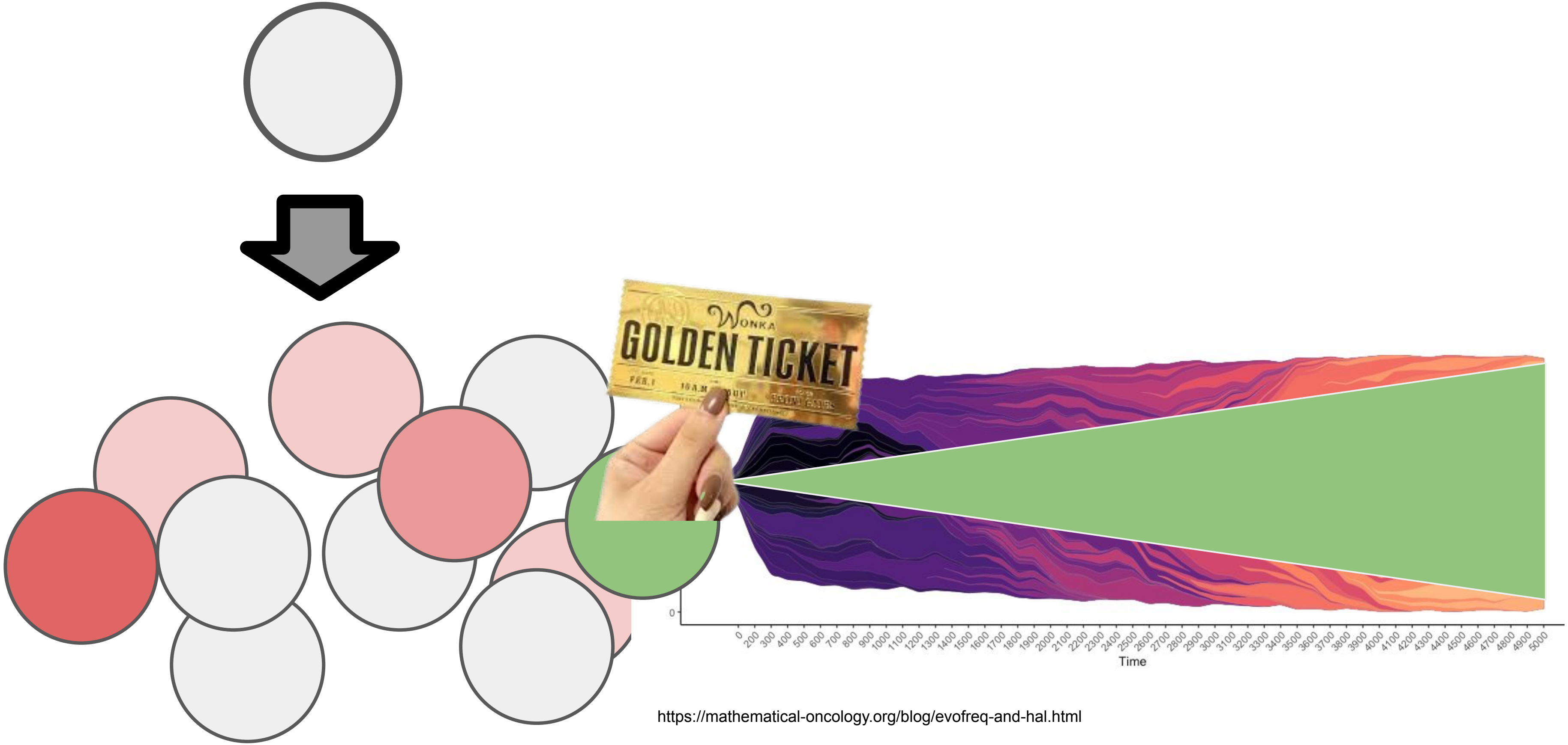
“hypermutator”



https://mathematical-oncology.org/blog/evofreq-and-hal.html

https://mathematical-oncology.org/blog/evofreq-and-hal.html

“hypermutator”



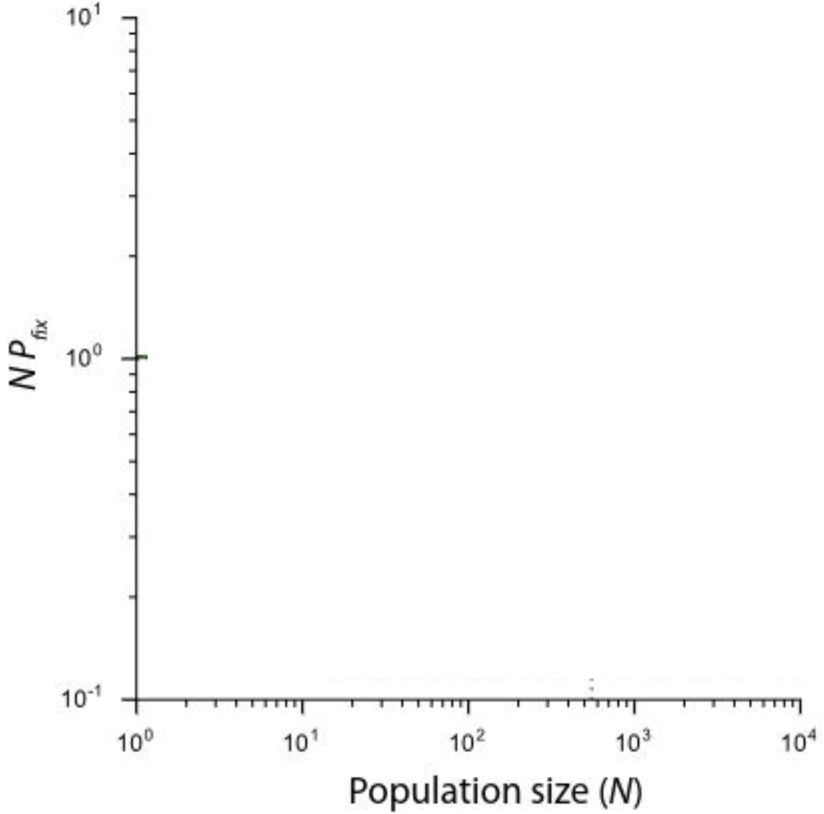
https://mathematical-oncology.org/blog/evofreq-and-hal.html

Hypermutator dynamics pertain to

- pathogen evolution
- antibiotic resistance
- tumor cell evolution
- etc.

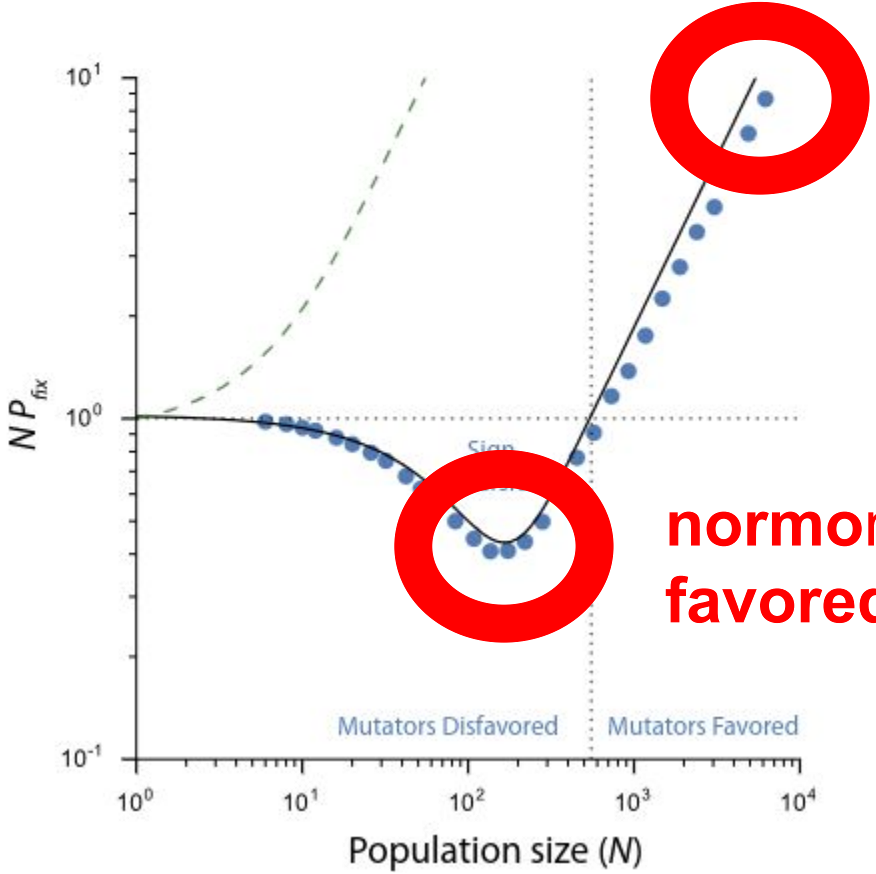
📧 morenoma@umich.edu

hypermutator selection — Raynes et al., 2018



📧 morenoma@umich.edu

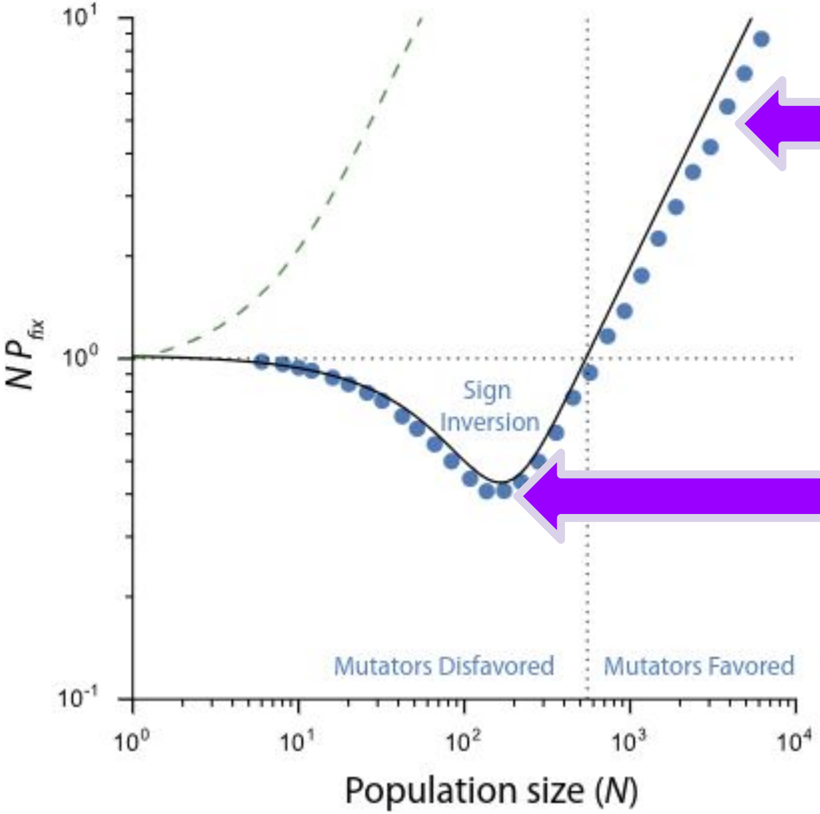
hypermutator selection — Raynes et al., 2018

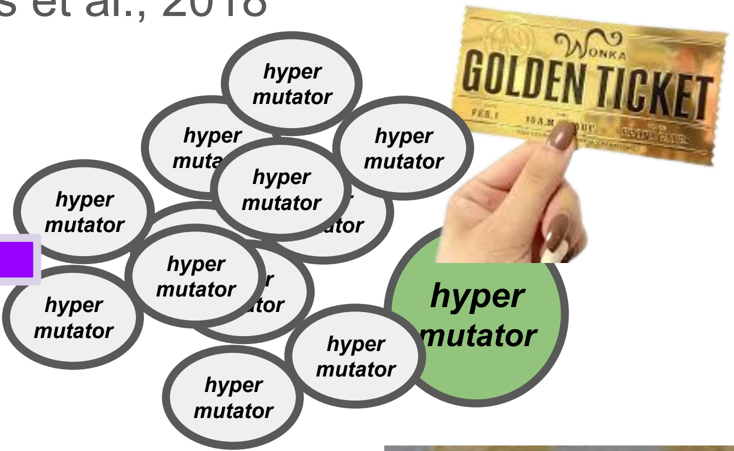


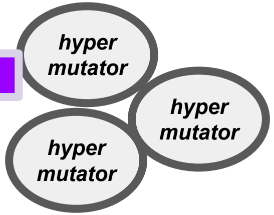
hypermutators
favored

normomutators
favored

hypermutator selection — Raynes et al., 2018

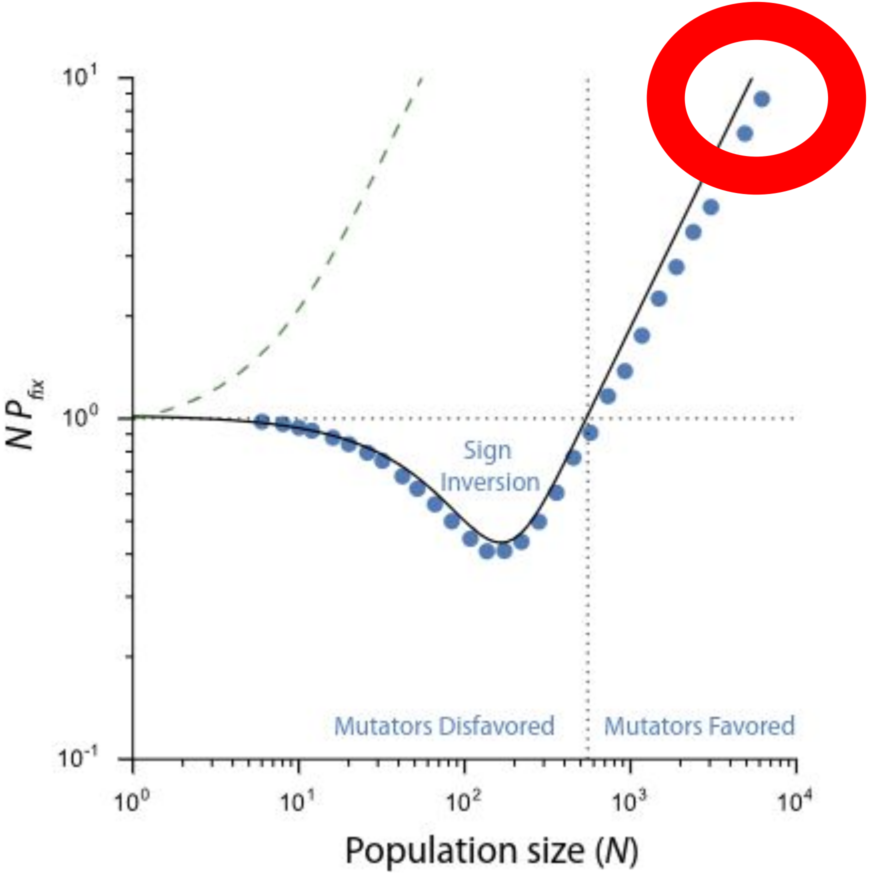






📧 morenoma@umich.edu

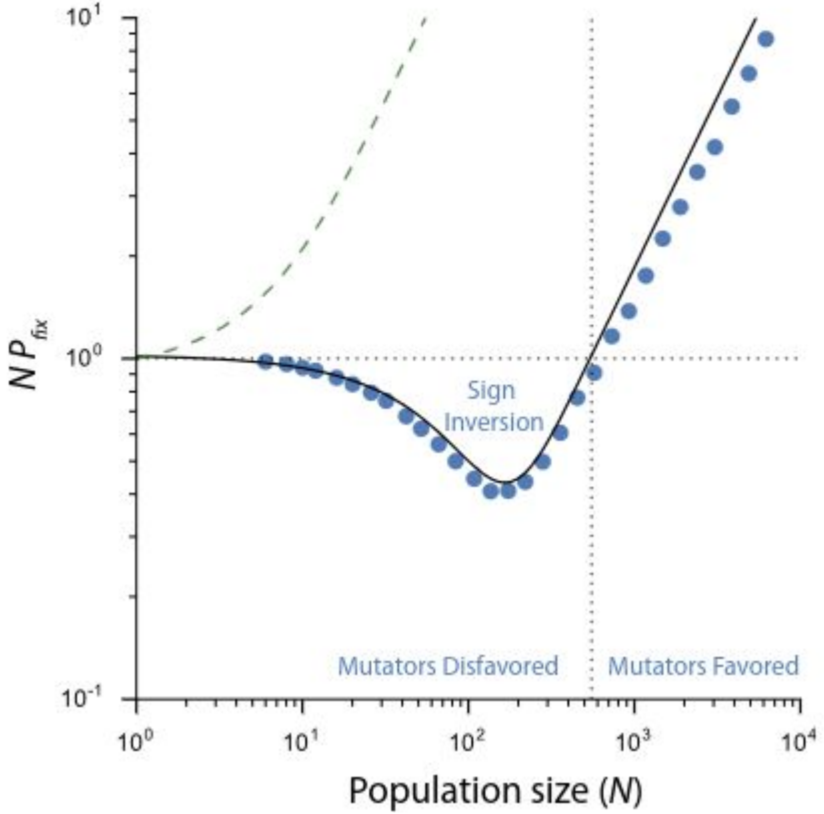
hypermutator selection — Raynes et al., 2018



hypermutators win

<— pop size 10k

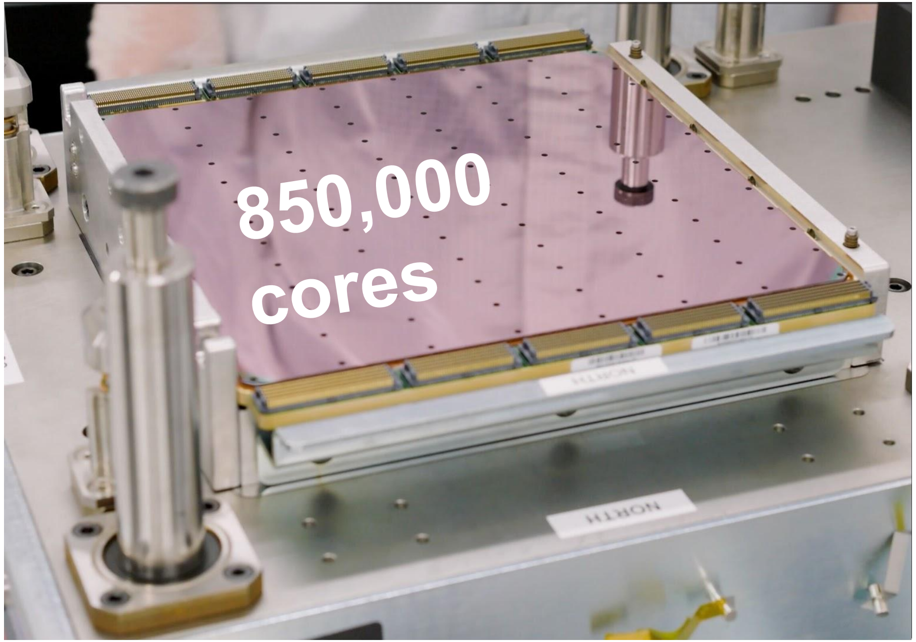
hypermutator selection — Raynes et al., 2018



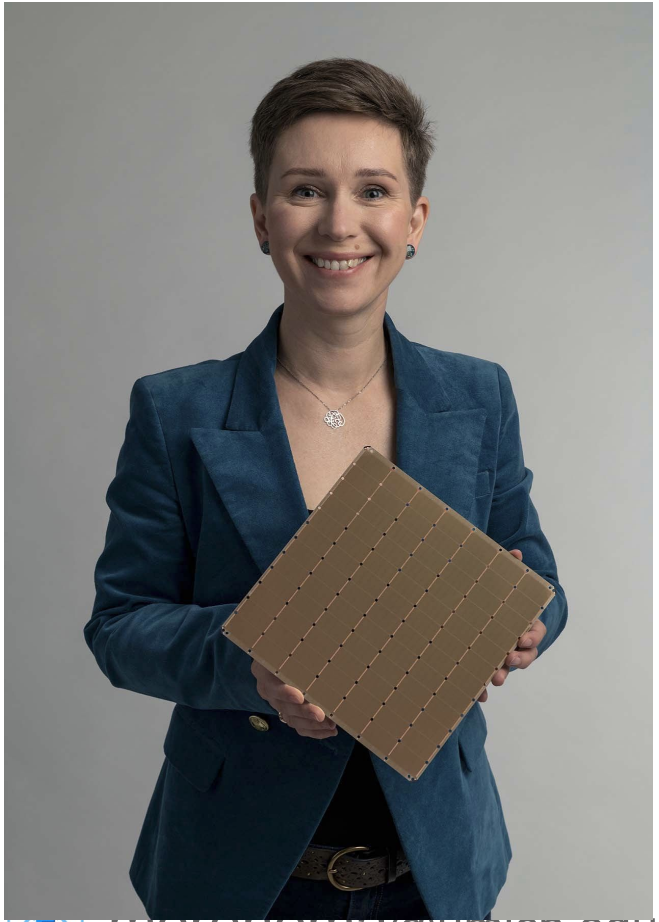
?????????

very large population sizes??

Cerebras Wafer-Scale Engine



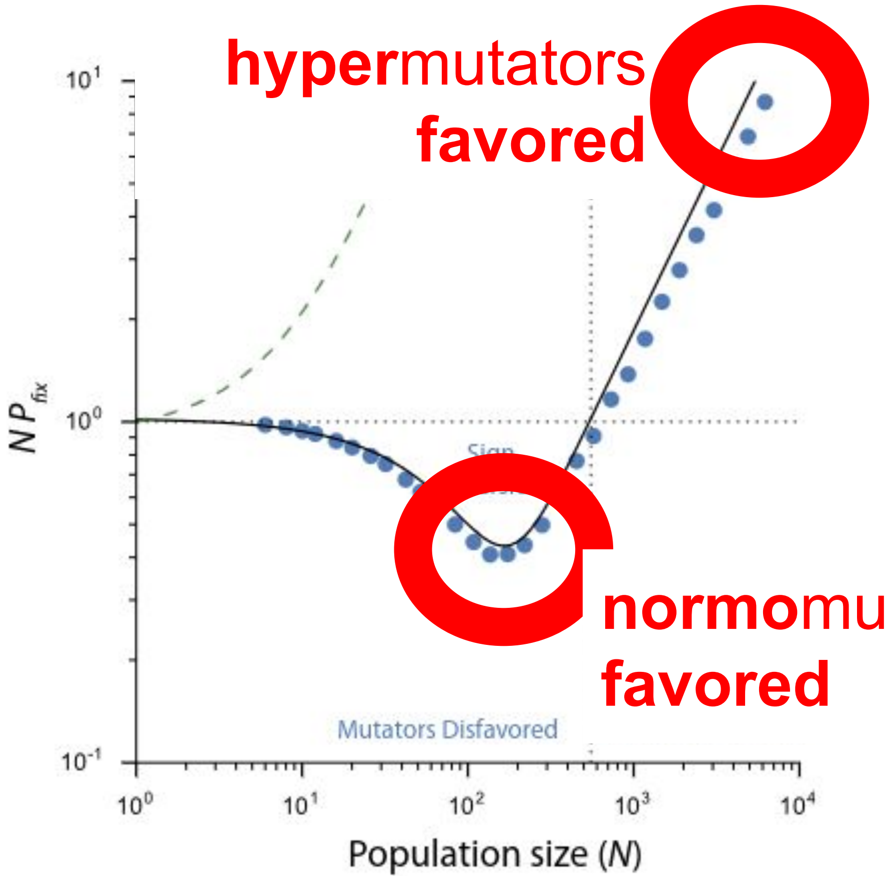
850,000 cores



📚 [these slides] https://hopth.ru/ea

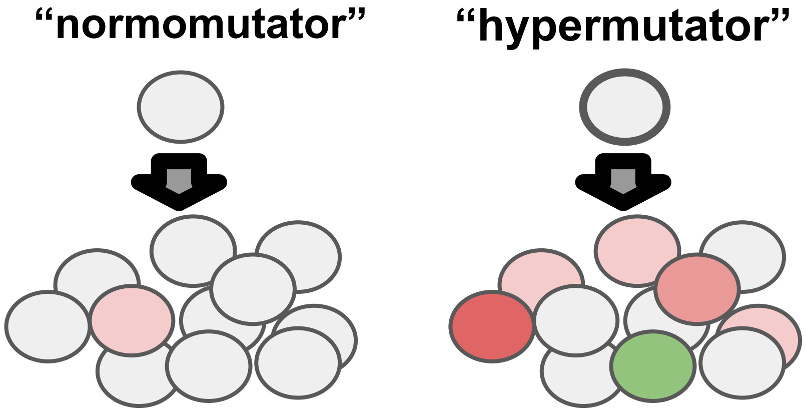
morenoma@umich.edu

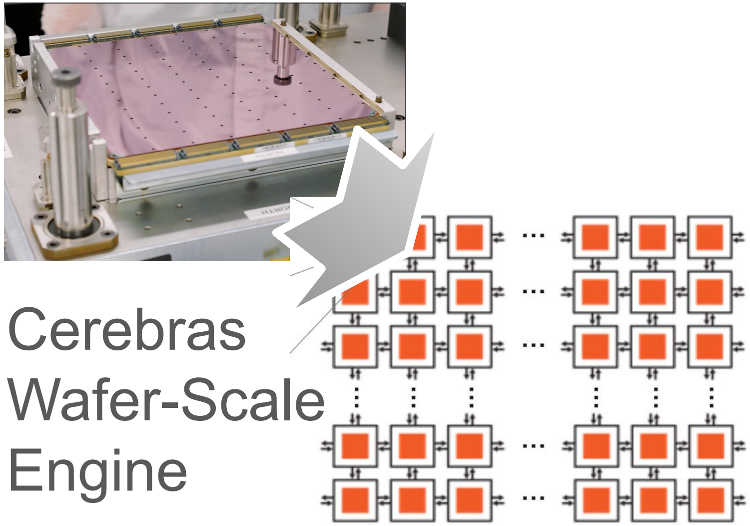
hypermutator selection — Raynes et al., 2018



hypermutators favored

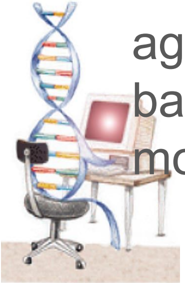
normomutators favored



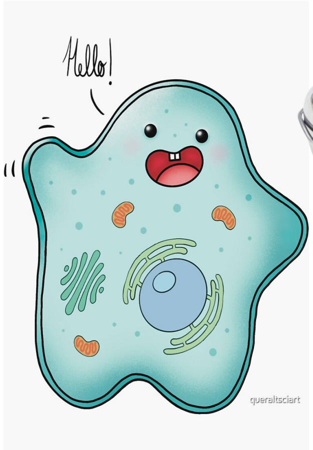


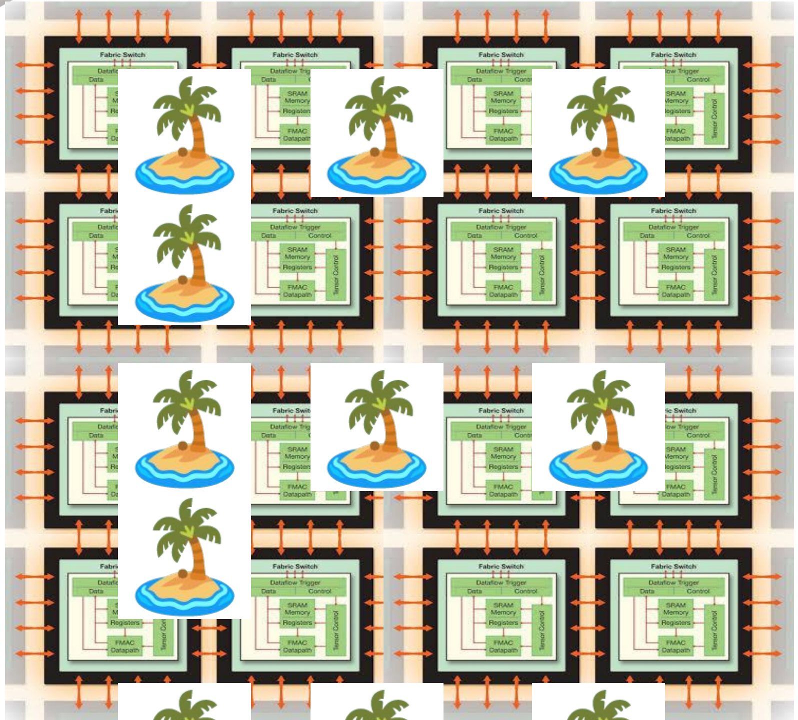
Cerebras Wafer-Scale Engine

📚 [these slides] https://hopth.ru/ea



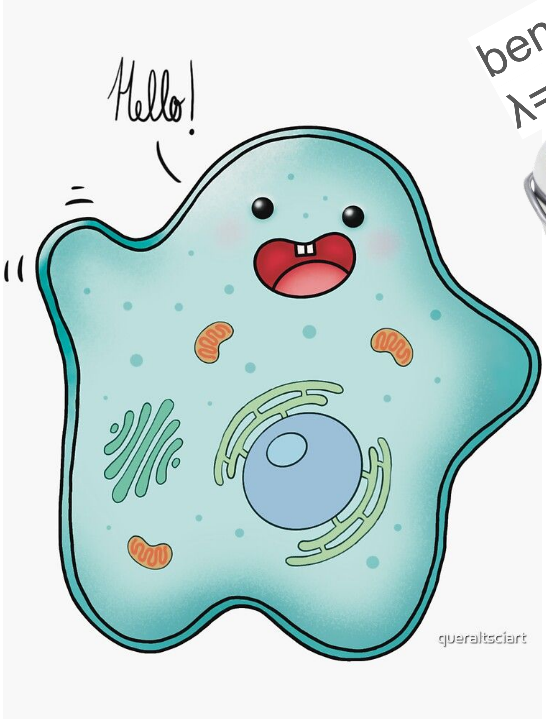
agent-based model





agent-based model

hypermutator selection — Raynes et al., 2018



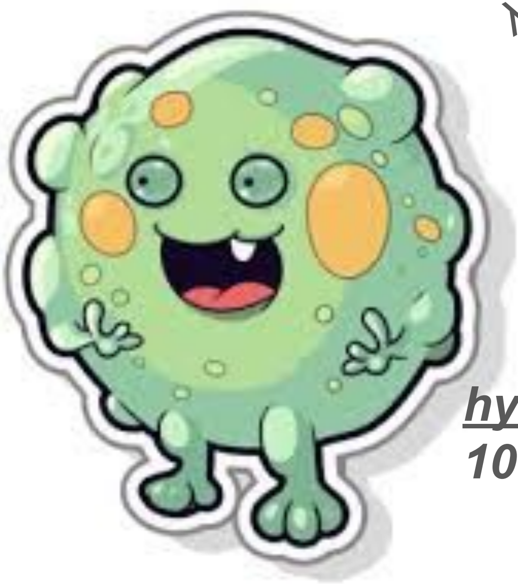
beneficial
λ=1 in 100k

deleterious
λ=1 in 1k

poisson distribution

agent-based
model

hypermutator selection — Raynes et al., 2018



hypermutator
100x mutation rate

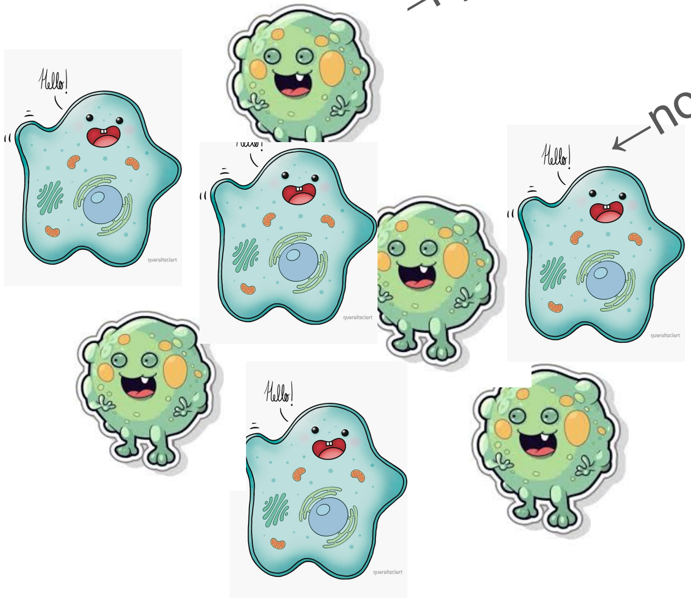
beneficial
λ=**1 in 1k**

deleterious
λ=**1 in 10**

poisson distribution

hypermutator selection — Raynes et al., 2018

50/50

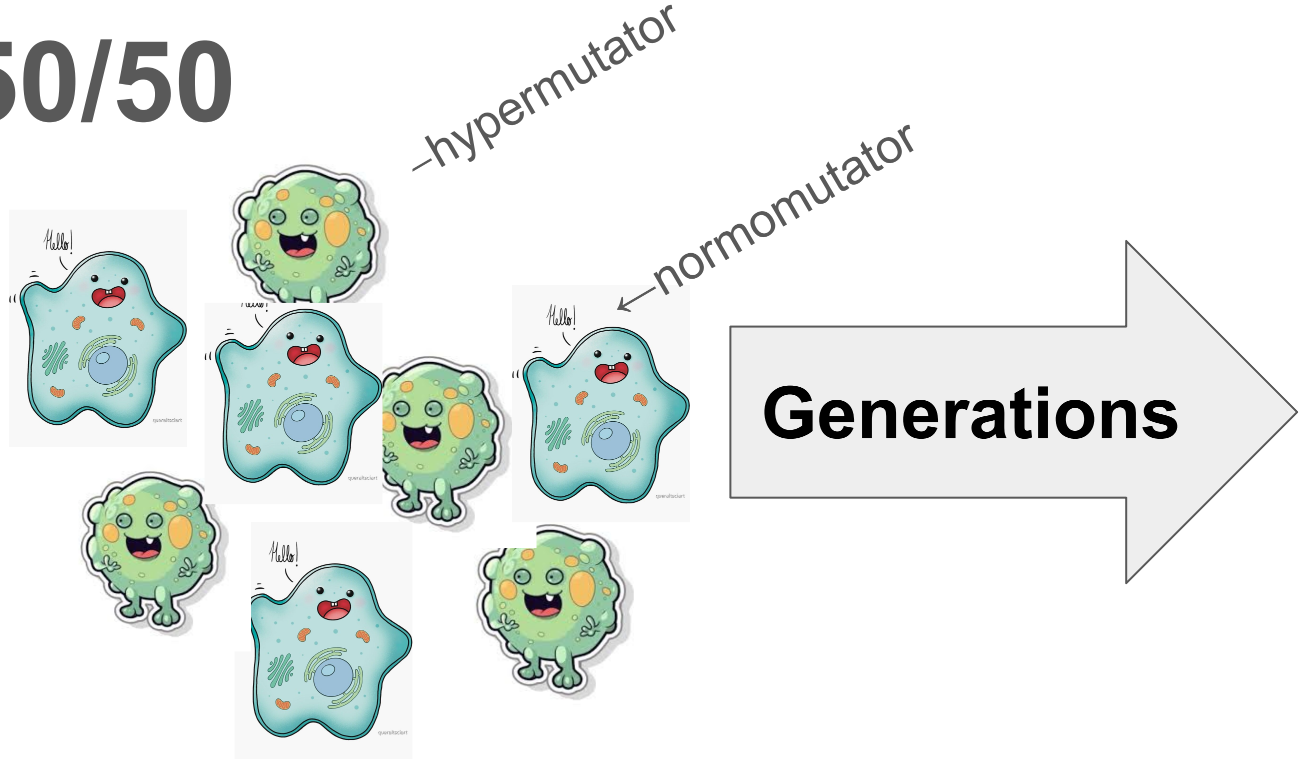


←hypermutator

←normomutator

hypermutator selection — Raynes et al., 2018

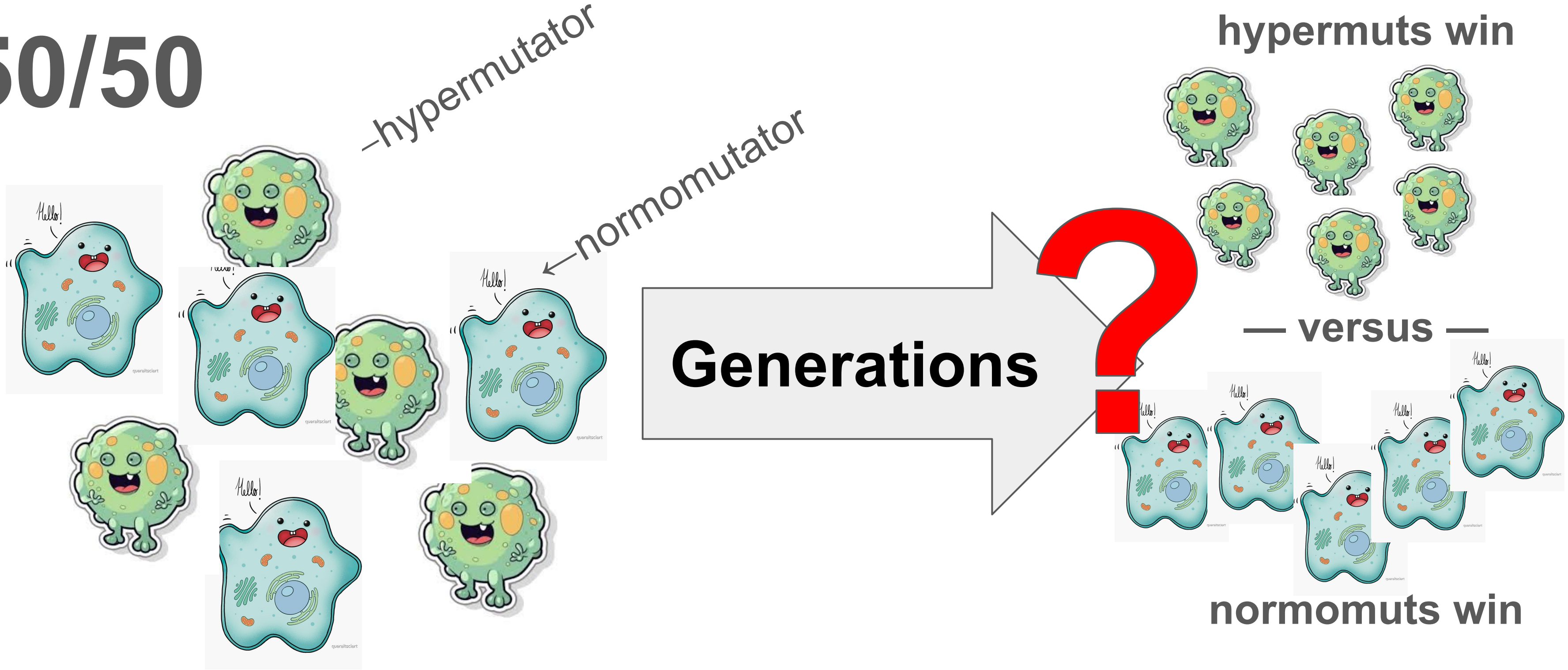
50/50



Generations

hypermutator selection — Raynes et al., 2018

50/50



–hypermutator

←normomutator

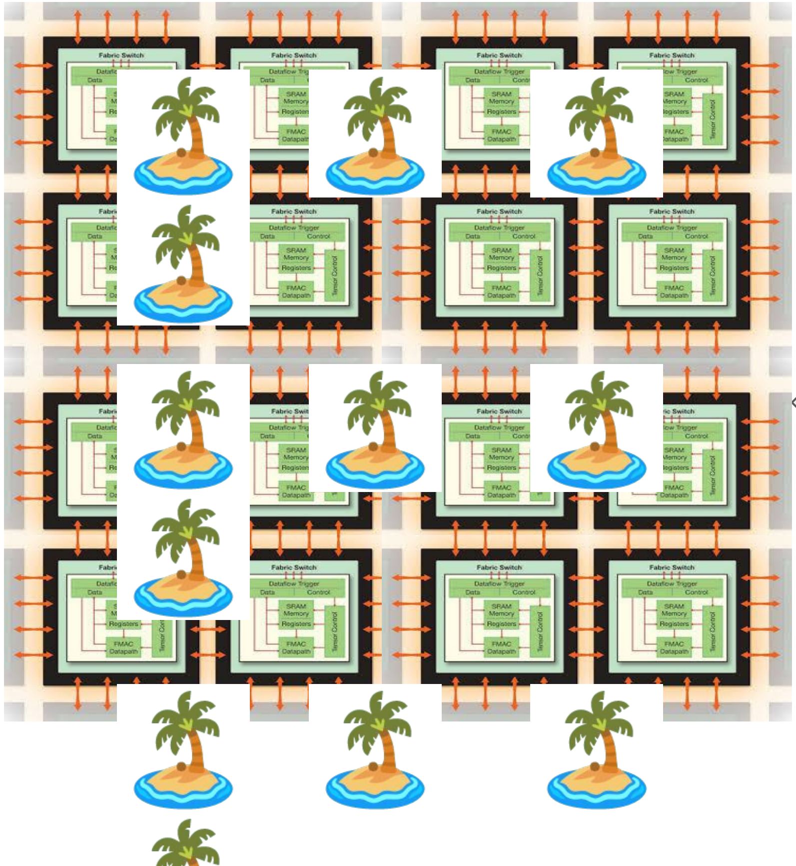
Generations

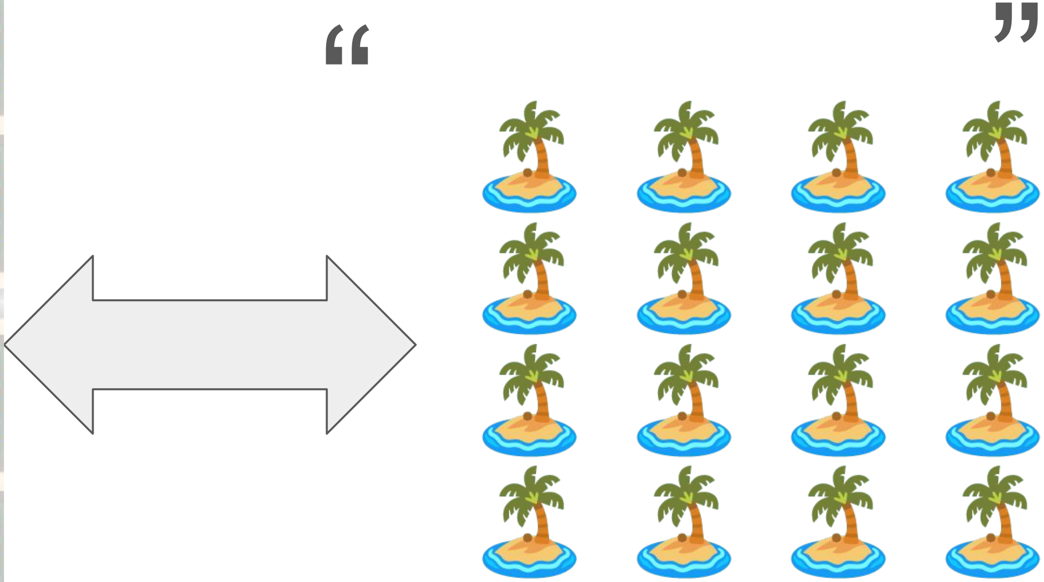
hypermuts win

— versus —

normomuts win

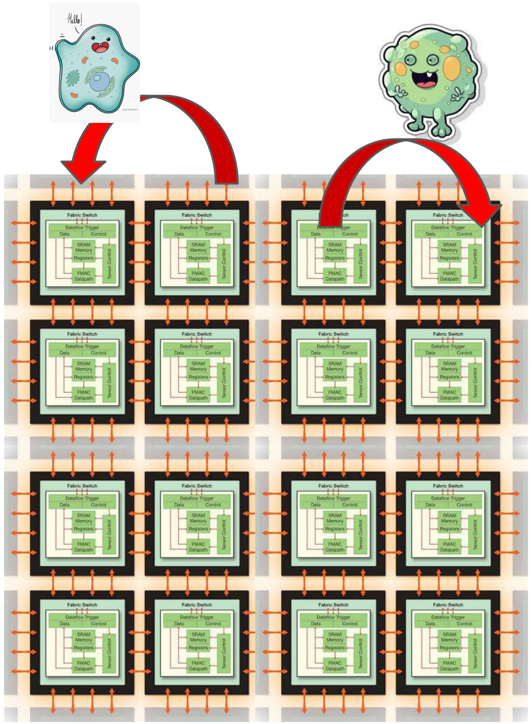
Island Model GA

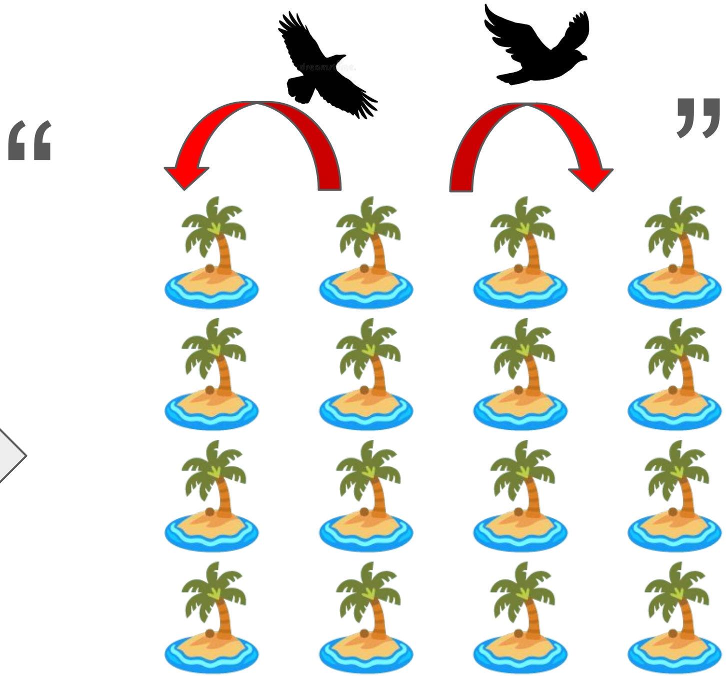




morenoma@umich.edu

Island Model GA

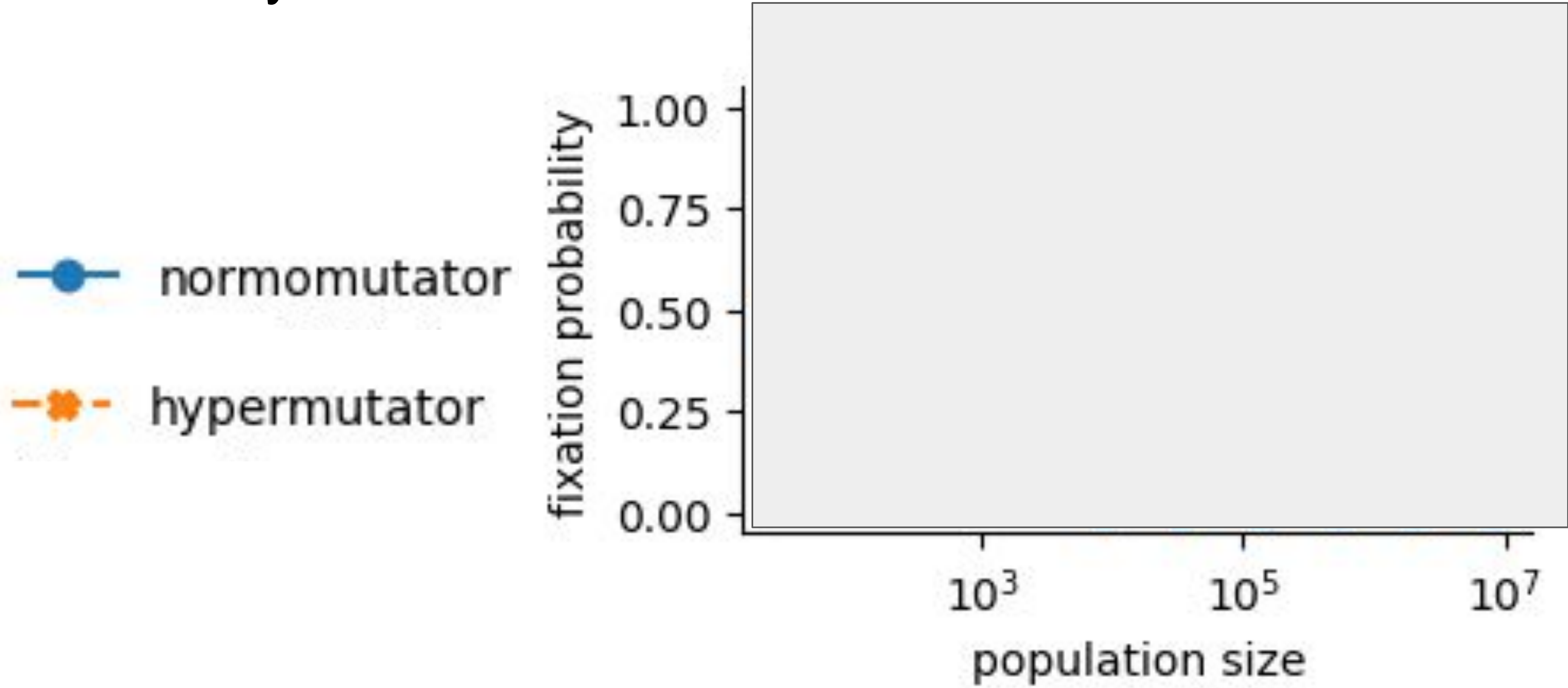




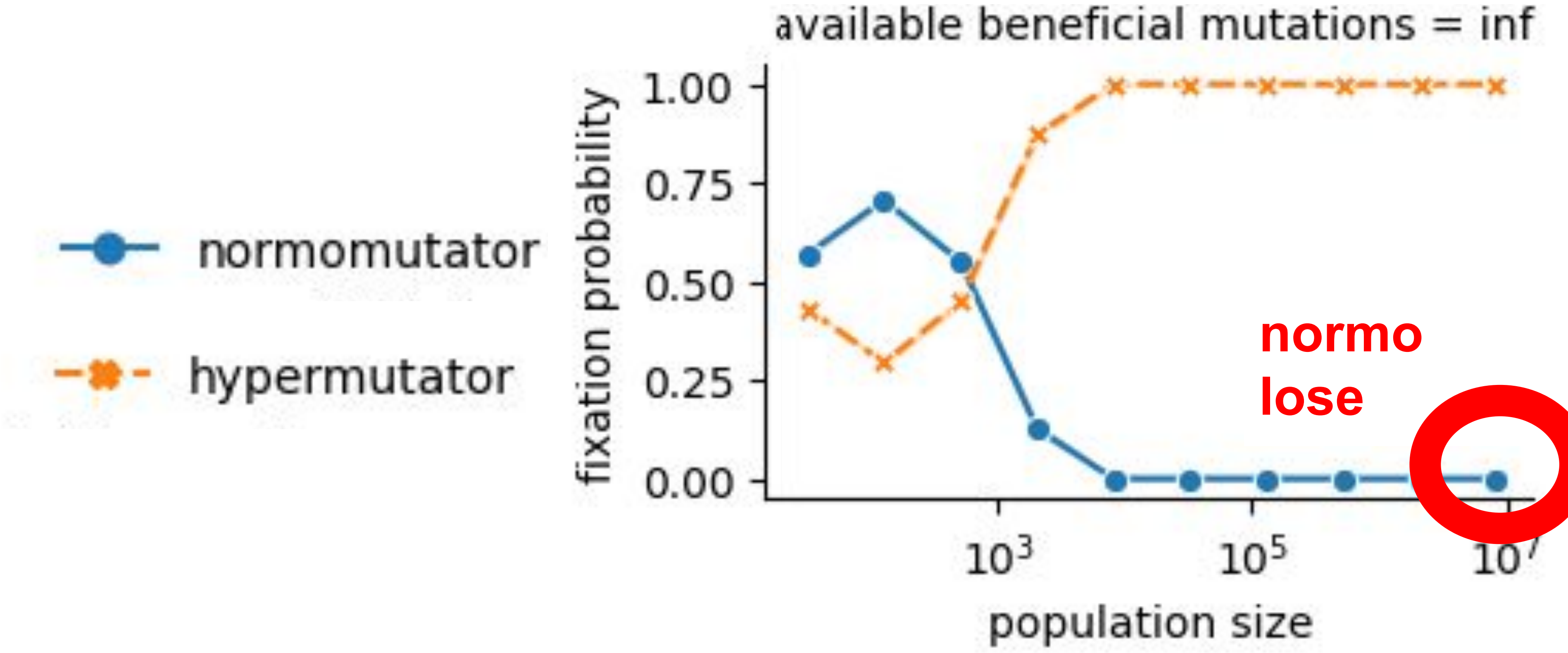
morenoma@umich.edu

Simulation Results

Preliminary Results



Preliminary Results

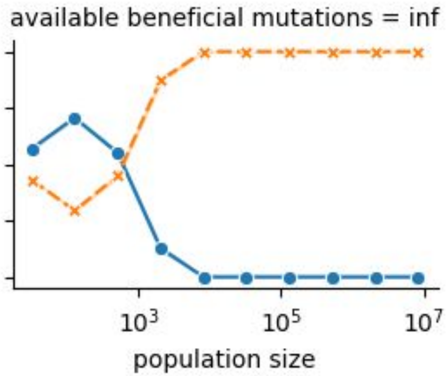


normo
lose

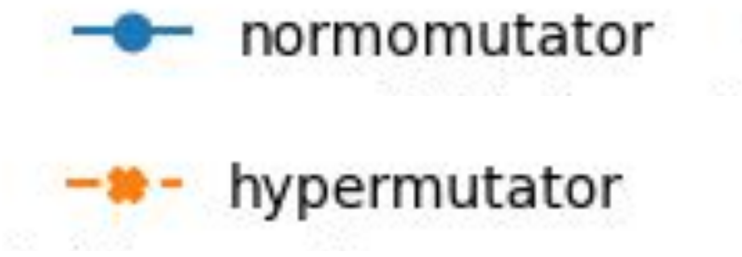
Preliminary Results

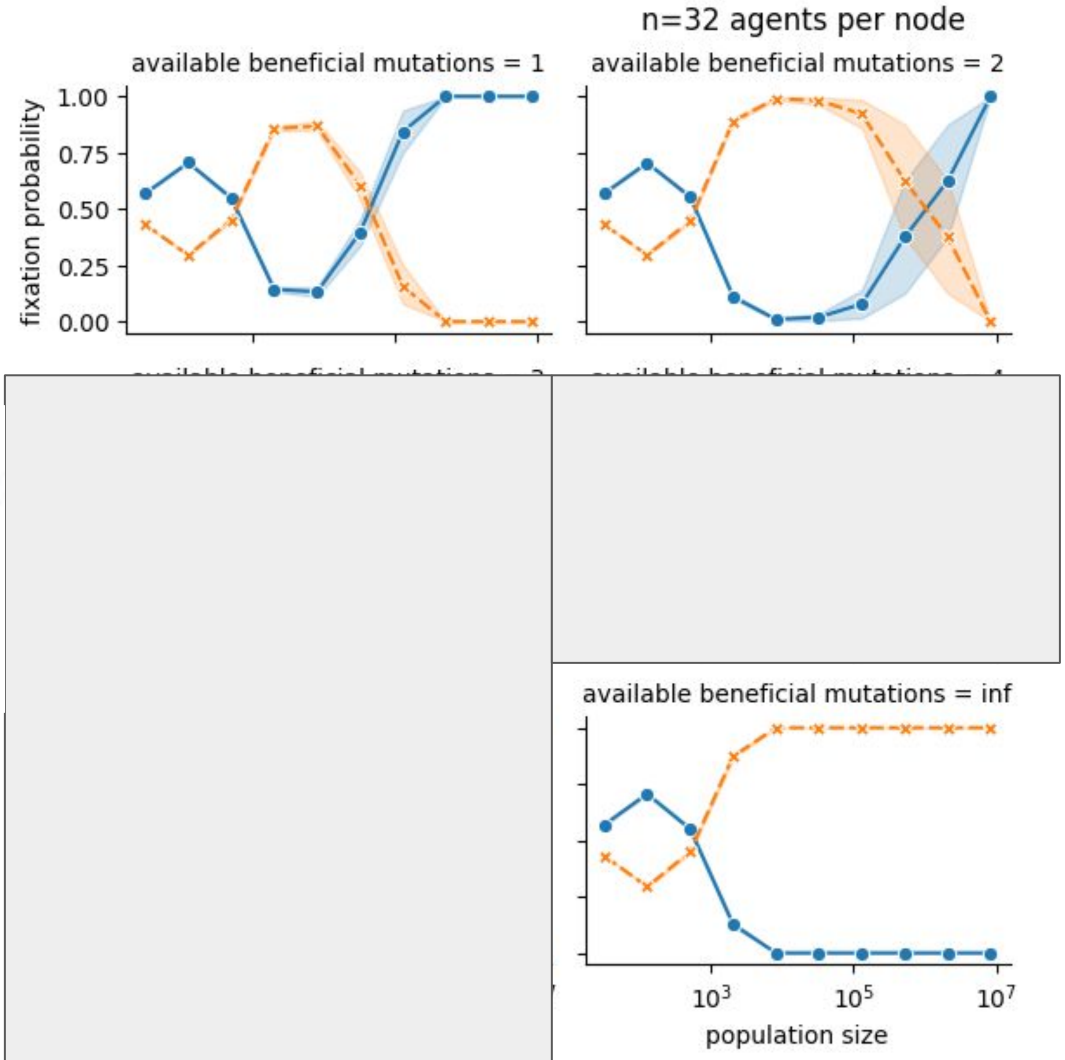
● normomutator

✕ hypermutator



Preliminary Results

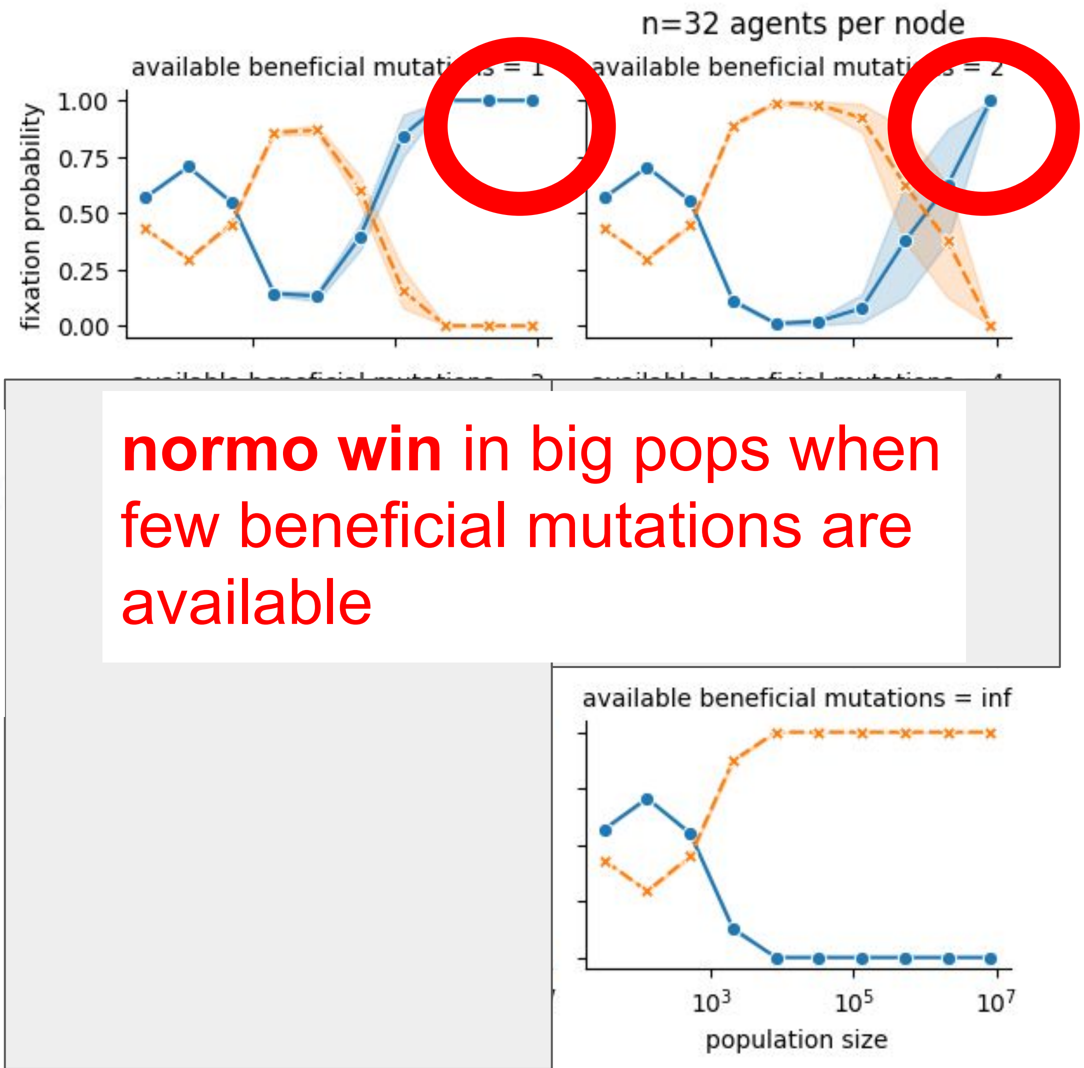




Preliminary Results

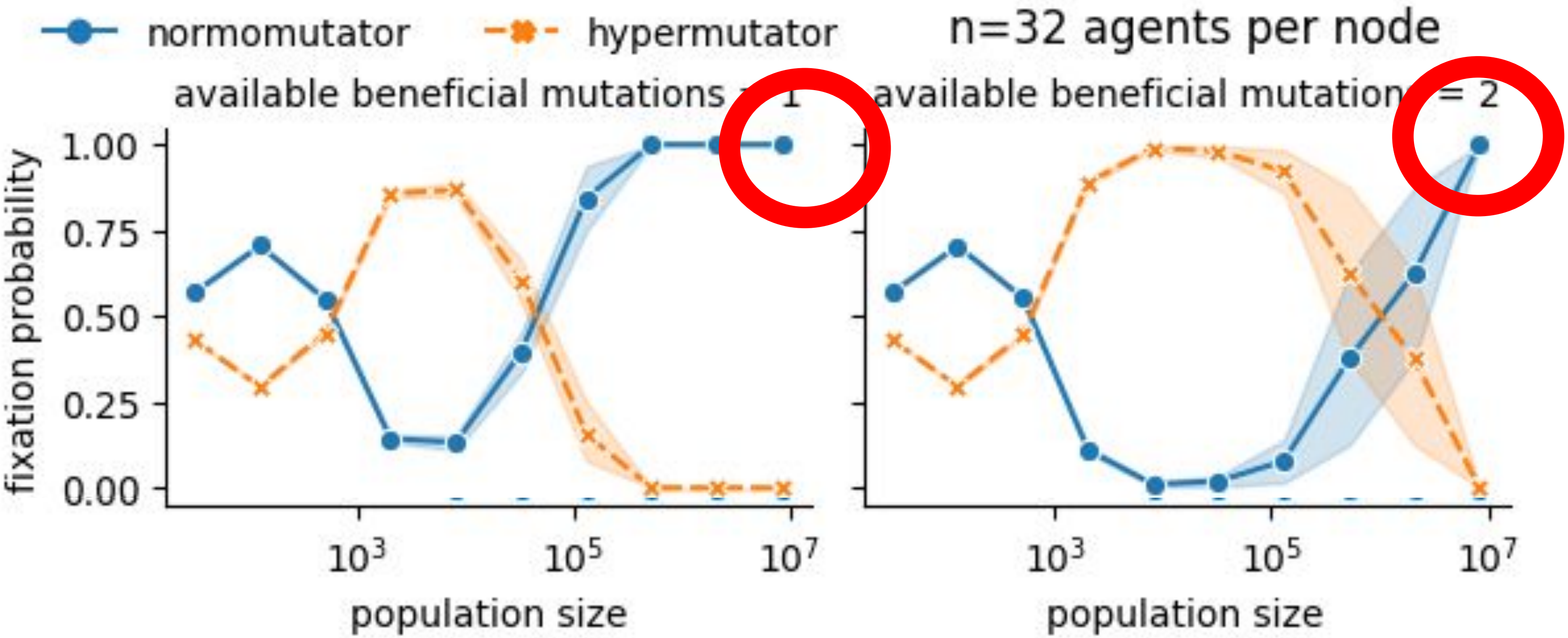
● normomutator

✖ hypermutator



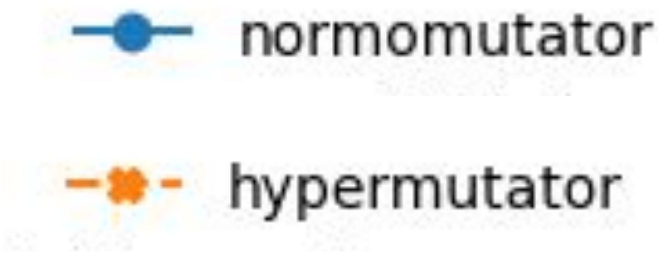
normo win in big pops when few beneficial mutations are available

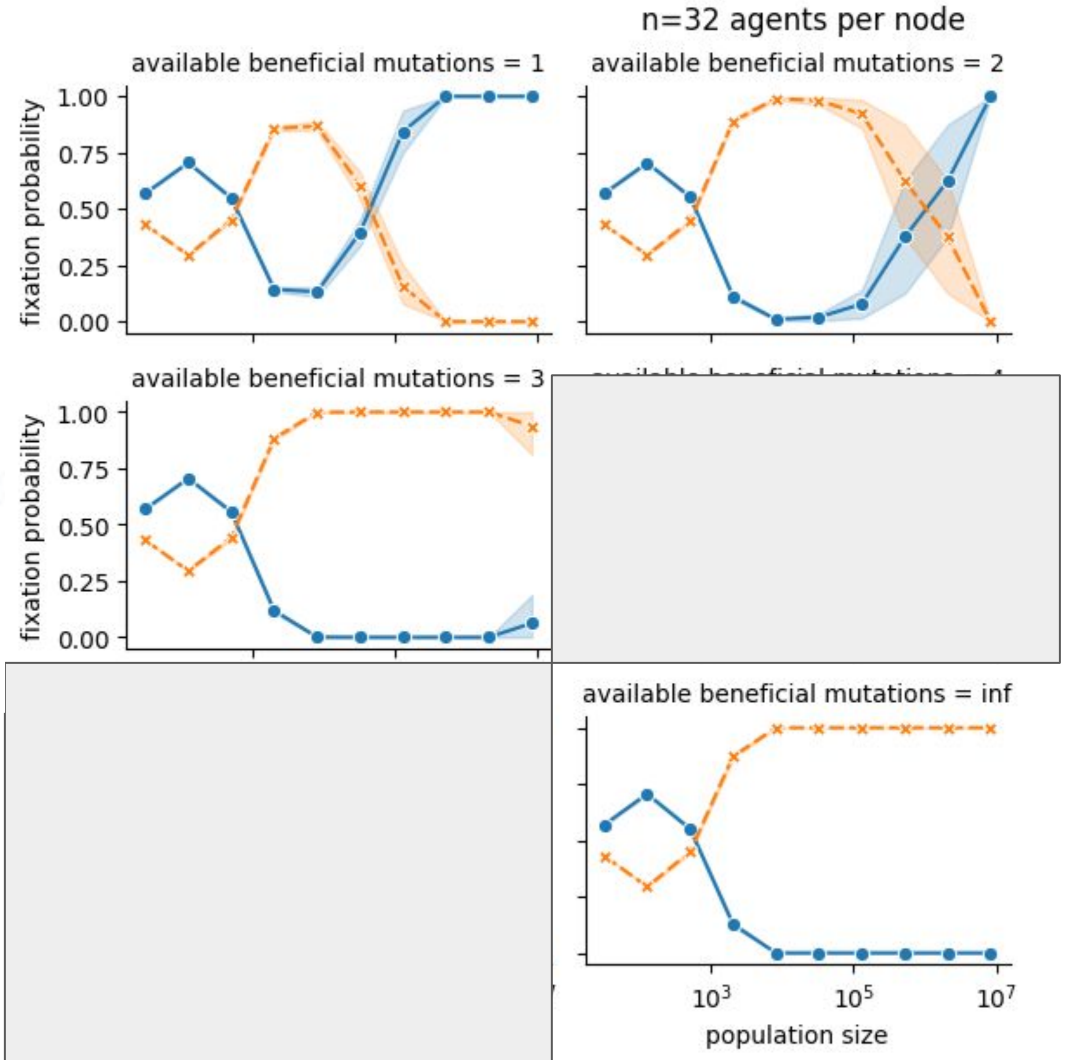
Preliminary Results



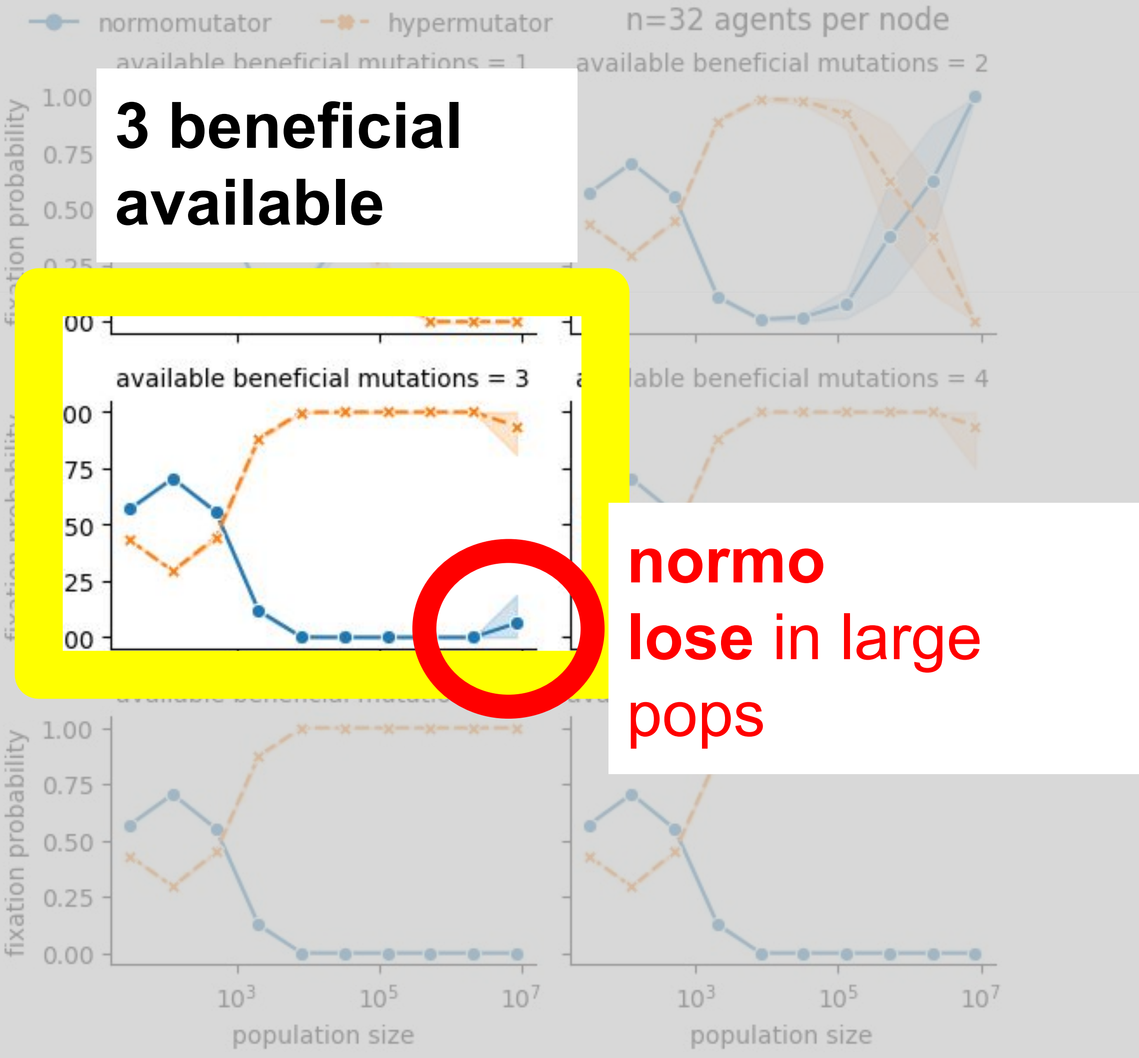
normo win in big pops when few beneficial mutations are available

Preliminary Results



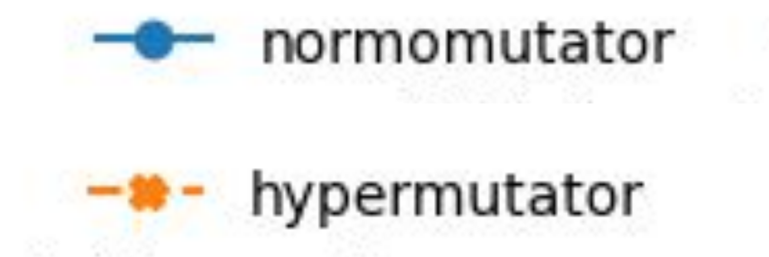


Preliminary Results



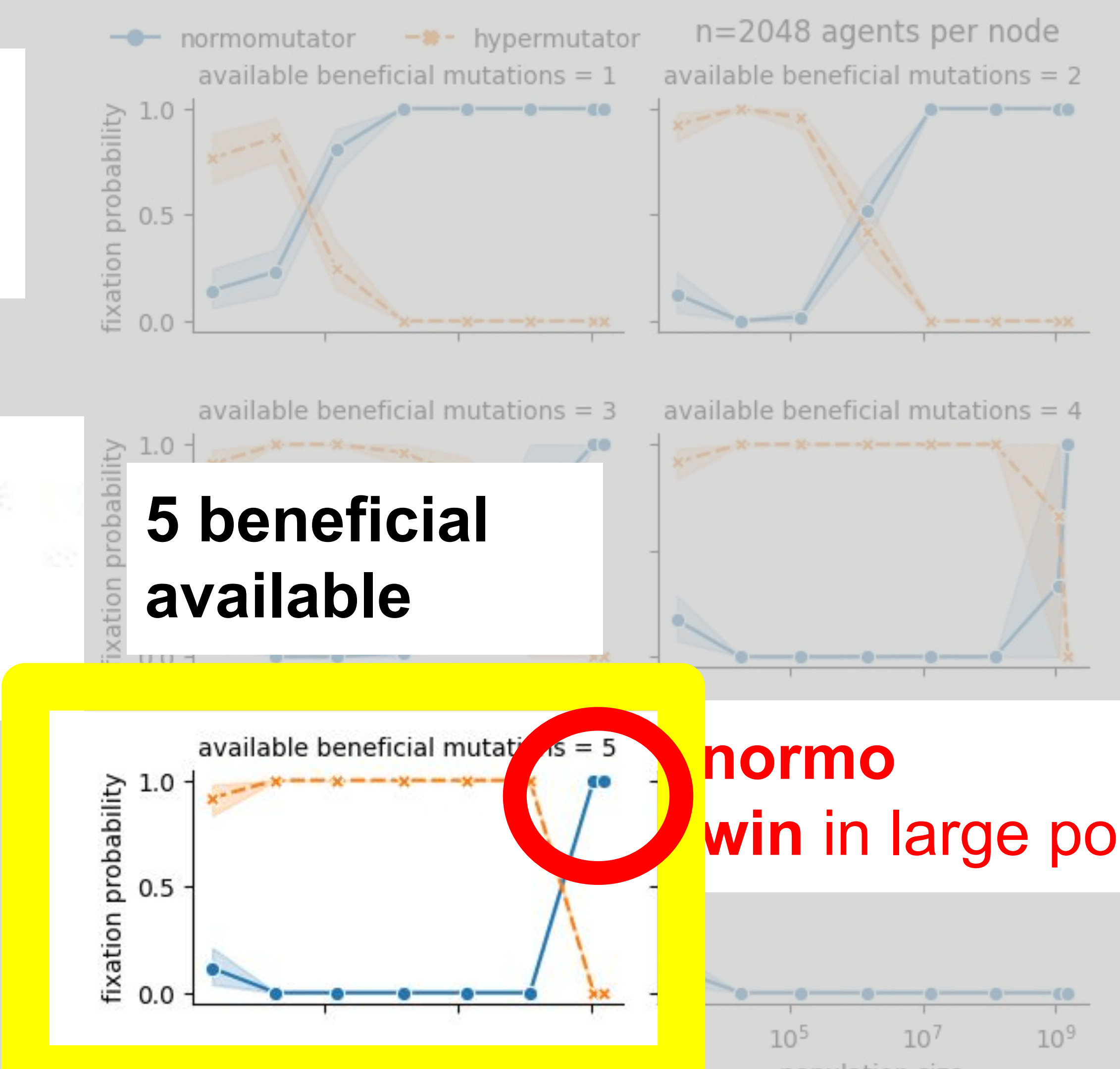
3 beneficial available

normo lose in large pops



pop size up to ~10 million

+very large pop size



● normomutator

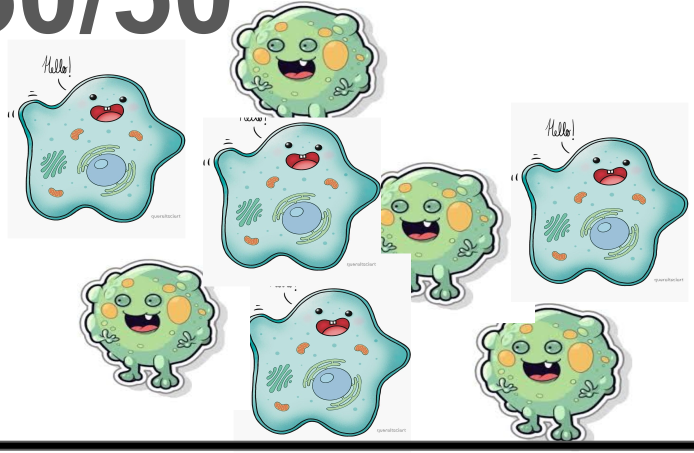
✖ hypermutator

5 beneficial available

normo win in large pops

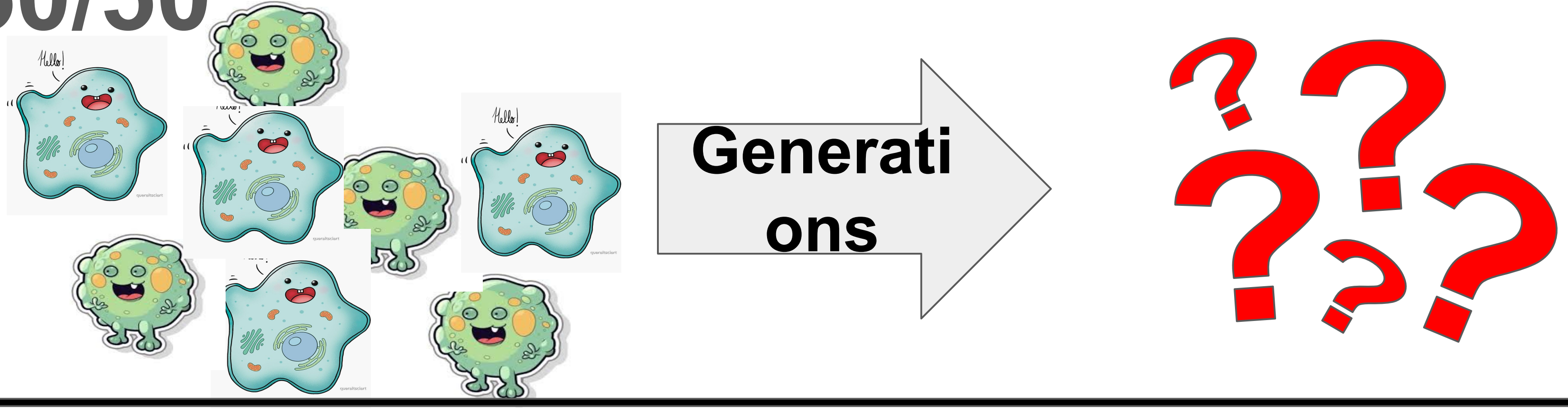
pop size up to ~***<u>1.5 billion</u>***

50/50



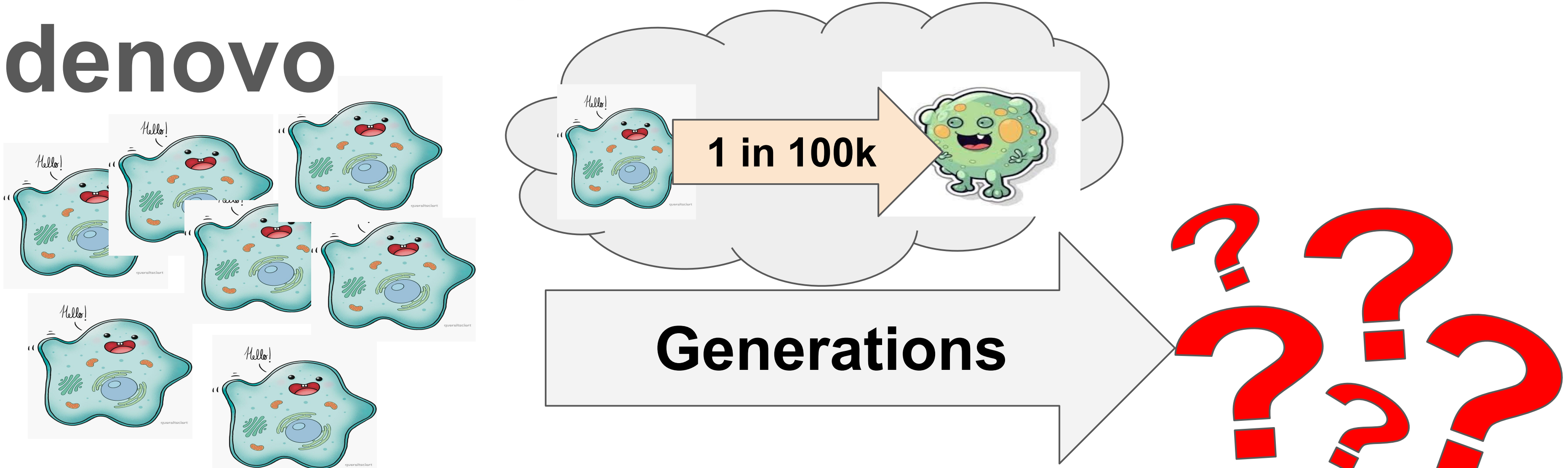
Generations

50/50



Generations

denovo

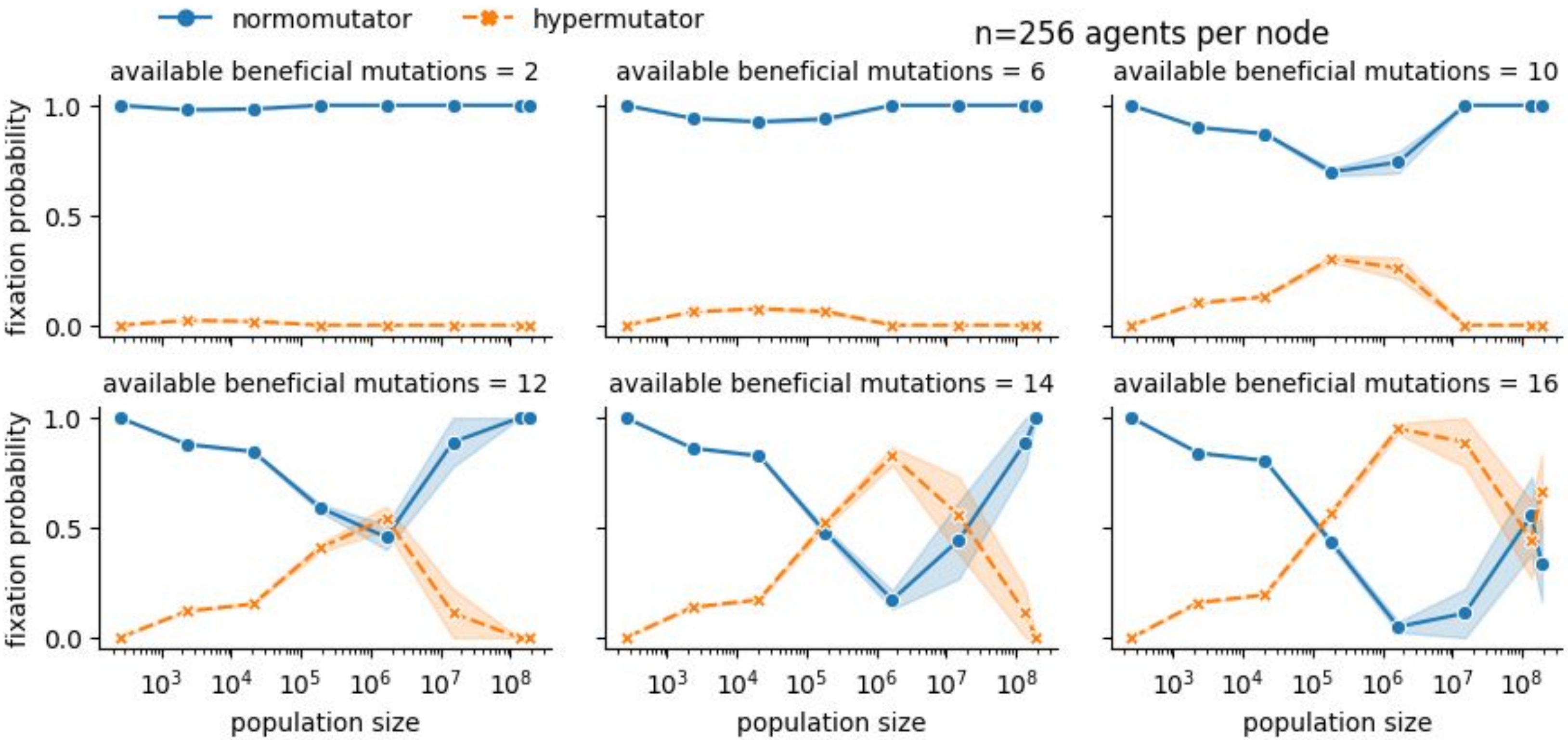


1 in 100k

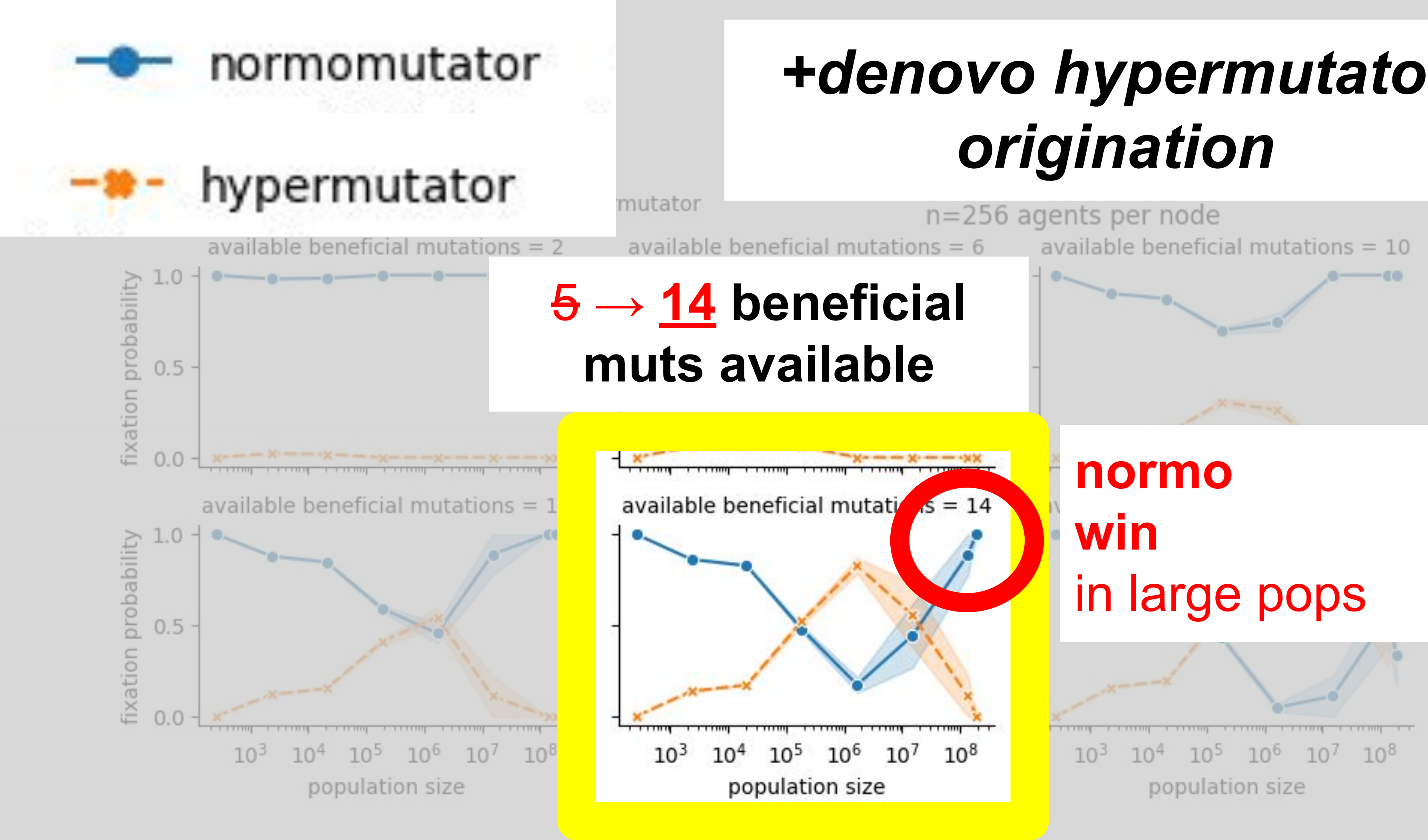
Generations

Preliminary Results

+denovo hypermutator origination



+denovo hypermutator origination



~~5~~ → 14 beneficial muts available

normo win

in large pops

on GPU

(*stronger*) ***spatial structure*** (*weaker*)

1D ring **2D grid** well-mixed

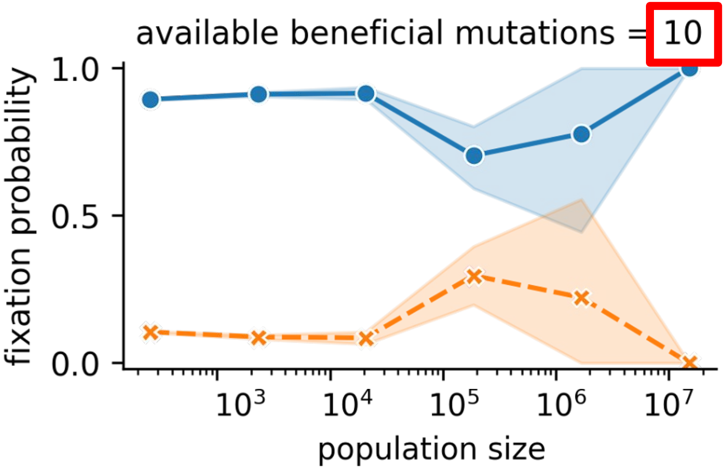
on GPU

(*stronger*) ***spatial structure*** (*weaker*)

1D ring

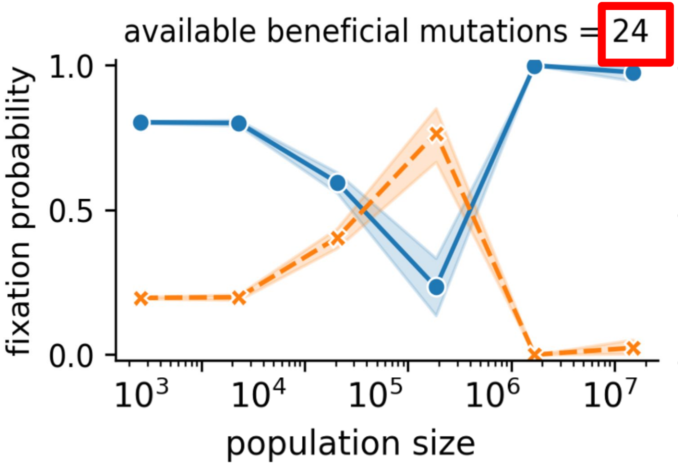
2D grid

well-mixed

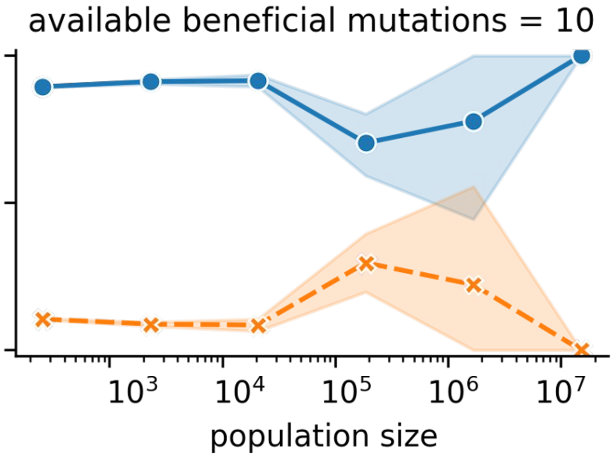


on GPU

1D ring



2D grid



well-mixed

normomutators **more** resilient

on GPU

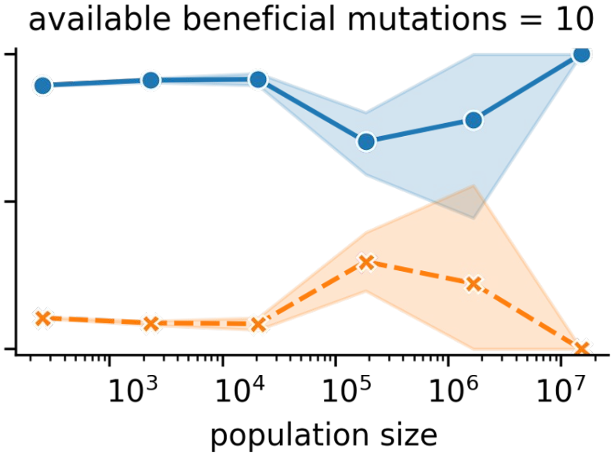
(stronger) **_spatial structure_** (weaker)

1D ring

2D grid

well-mixed

normomutators
more resilient
(>20 ben muts avail)



on GPU

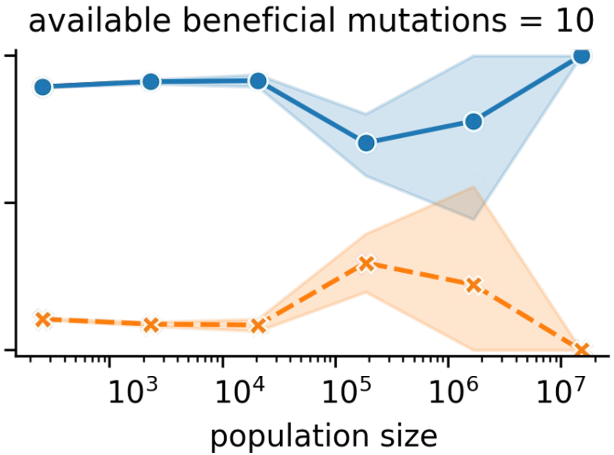
(stronger) ***spatial structure*** *(weaker)*

1D ring

2D grid

well-mixed

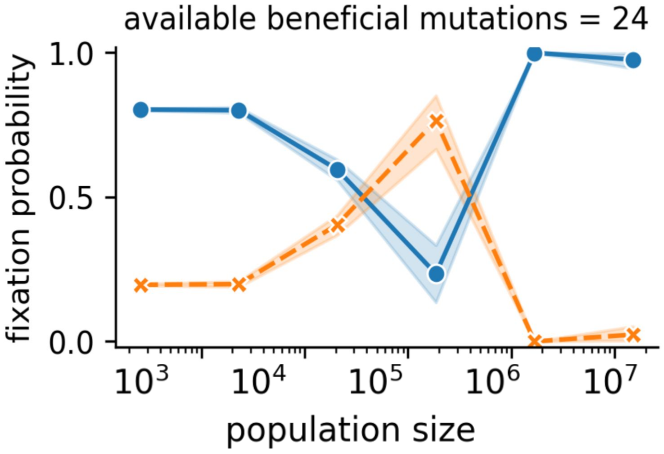
normomutators
more resilient
(>20 ben muts avail)



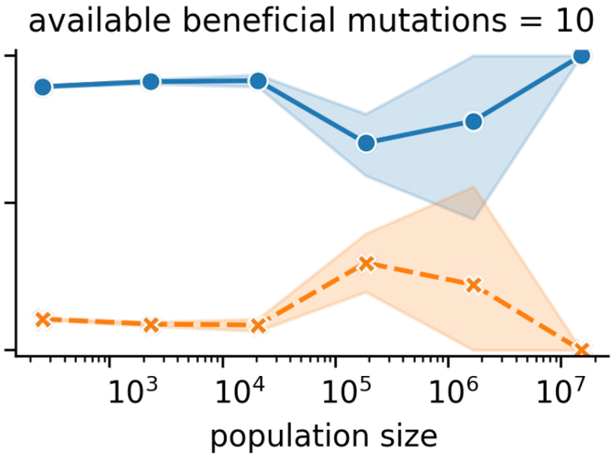
normomutators
less resilient
(<8 ben muts avail)

on GPU

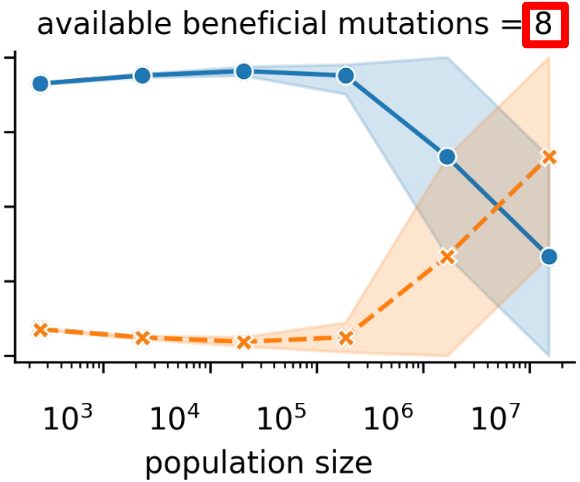
1D ring



2D grid



well-mixed



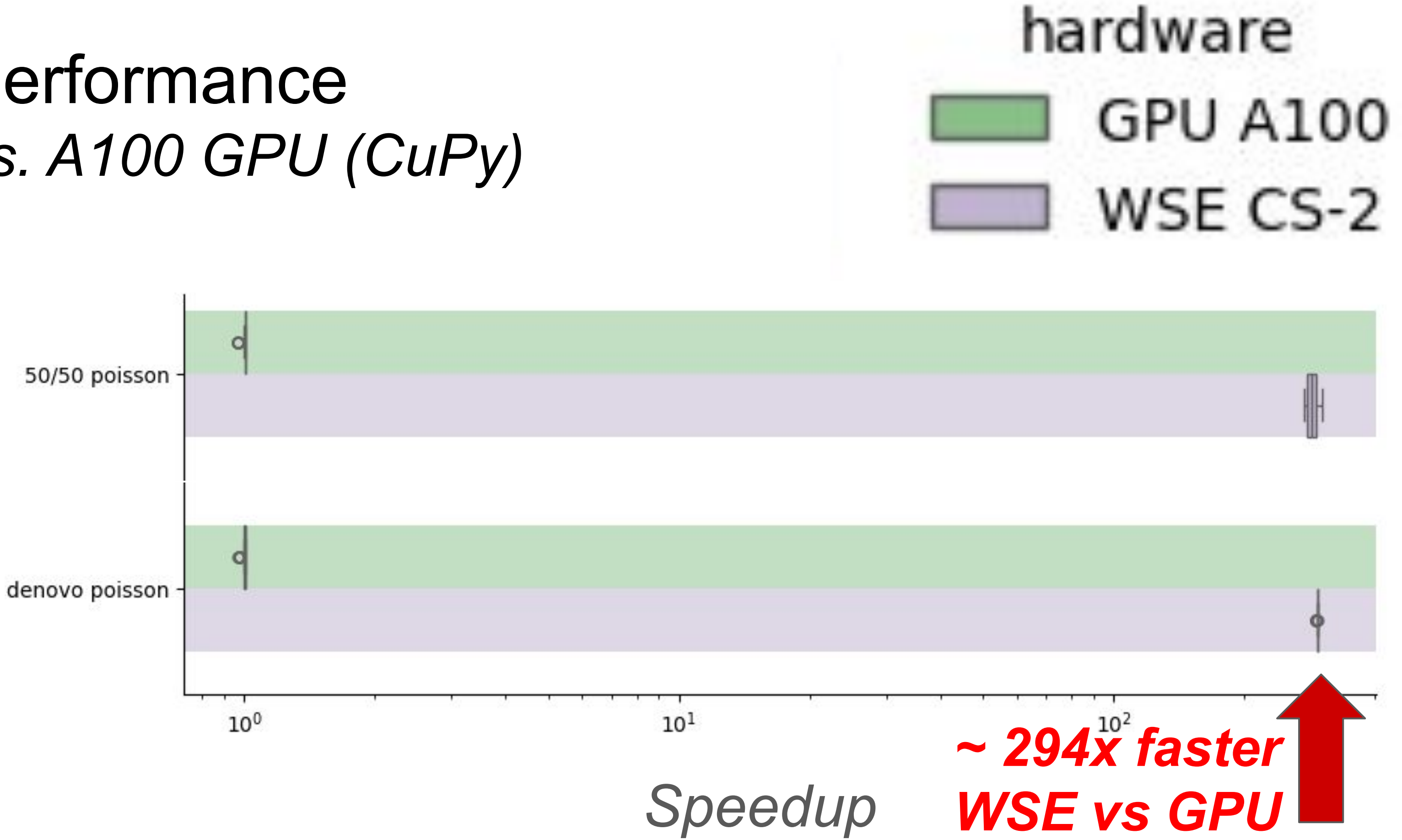
normomutators **more** resilient

normomutators **less** resilient

Performance 🏎️

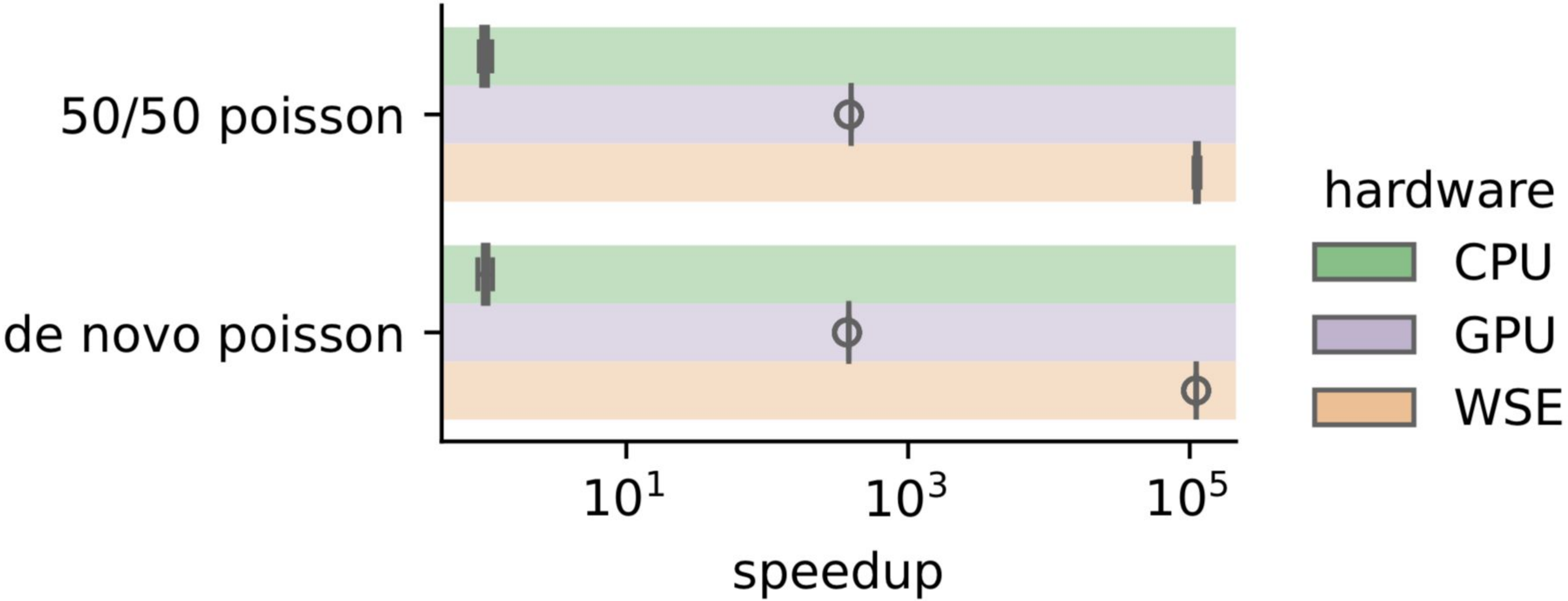
Performance

vs. A100 GPU (CuPy)

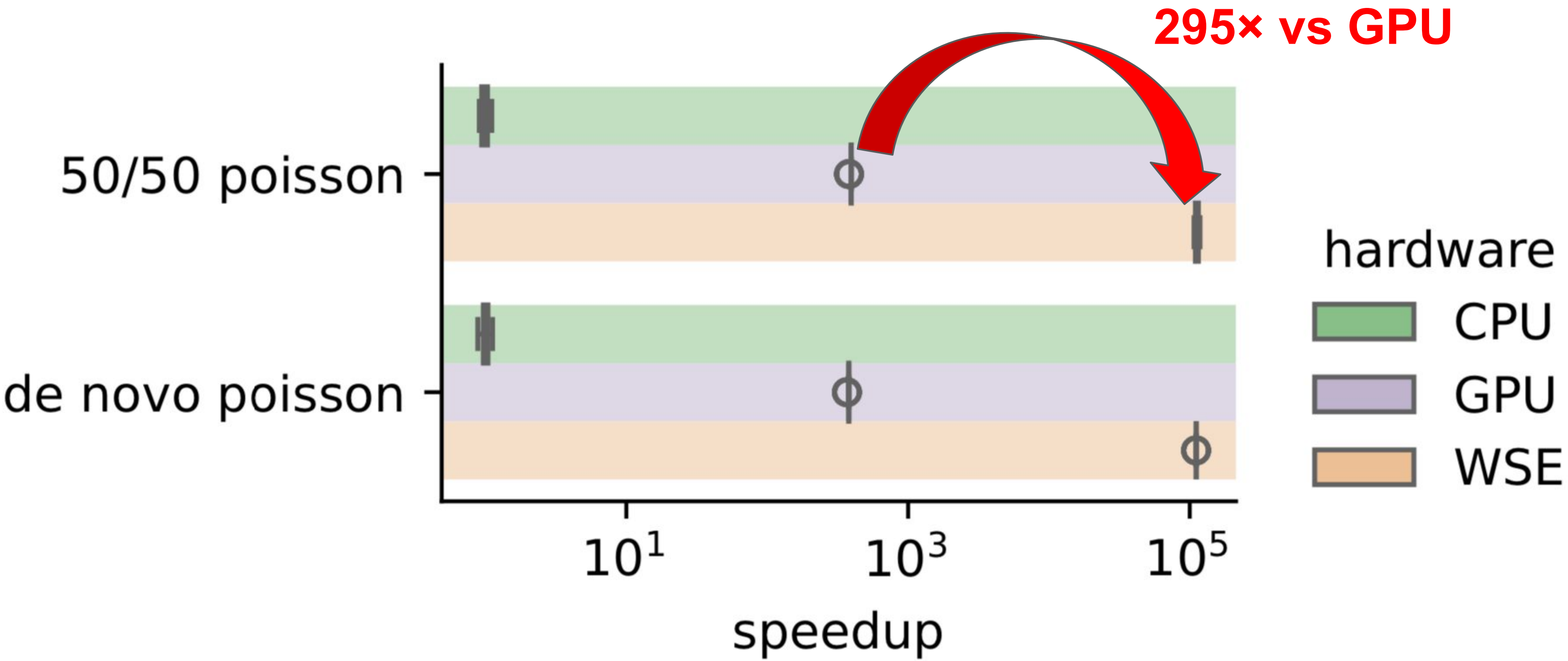


~ 294x faster WSE vs GPU

WSE ~8x10^{11} *(800 billion)* **agent-generations per second**



WSE ~8x10^{11} *(800 billion)* **agent-generations per second**



295× vs GPU

WSE ~8x10^{11} *(800 billion)* **agent-generations per second**

Performance

Agents per PE	Net Population Size	Simulation Speed (Generations per Second)

Performance

Agents per PE	Net Population Size	Simulation Speed (Generations per Second)
256	190.8 million	4,306 (std 156)

Performance

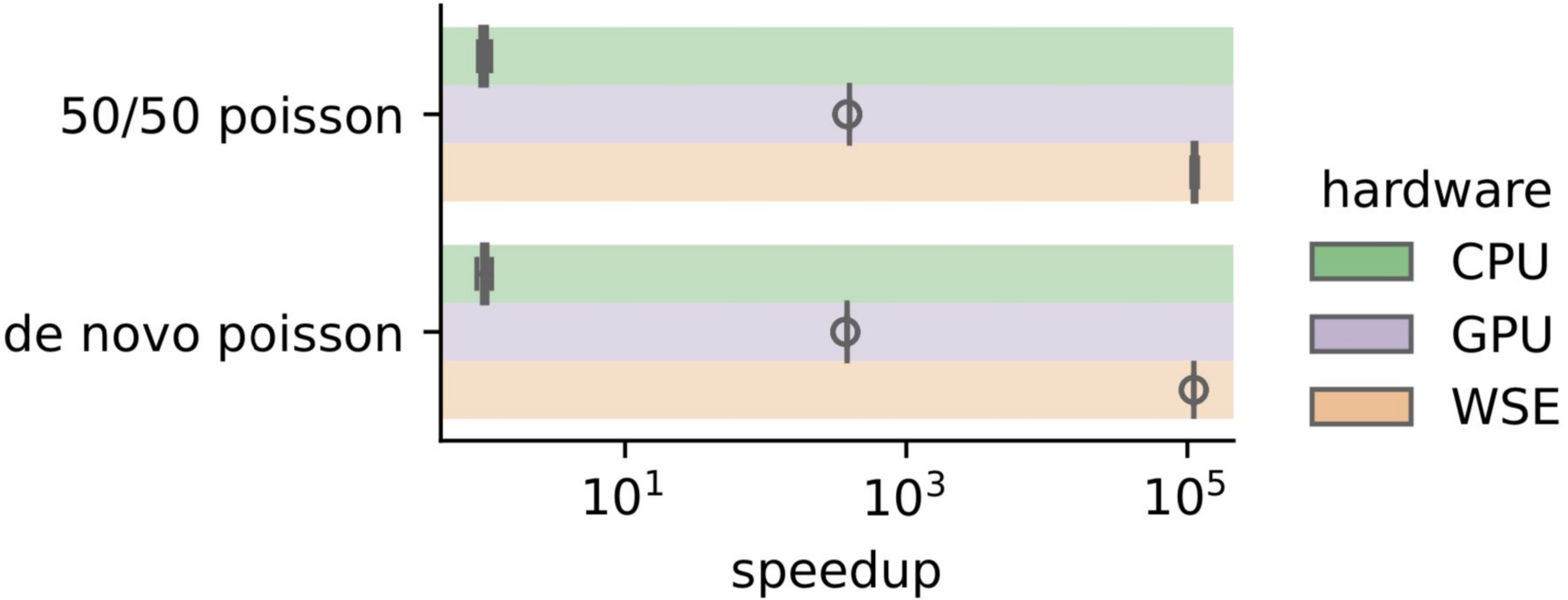
Agents per PE	**Net Population Size**	**Simulation Speed (Generations per Second)**
256	190.8 million	4,306 (std 156)
2048	1.5 billion	549 (std 1)

Performance

Agents per PE	**Net Population Size**	**Simulation Speed (Generations per Second)**
256	190.8 million	4,306 (std 156)
2048	1.5 billion	549 (std 1)

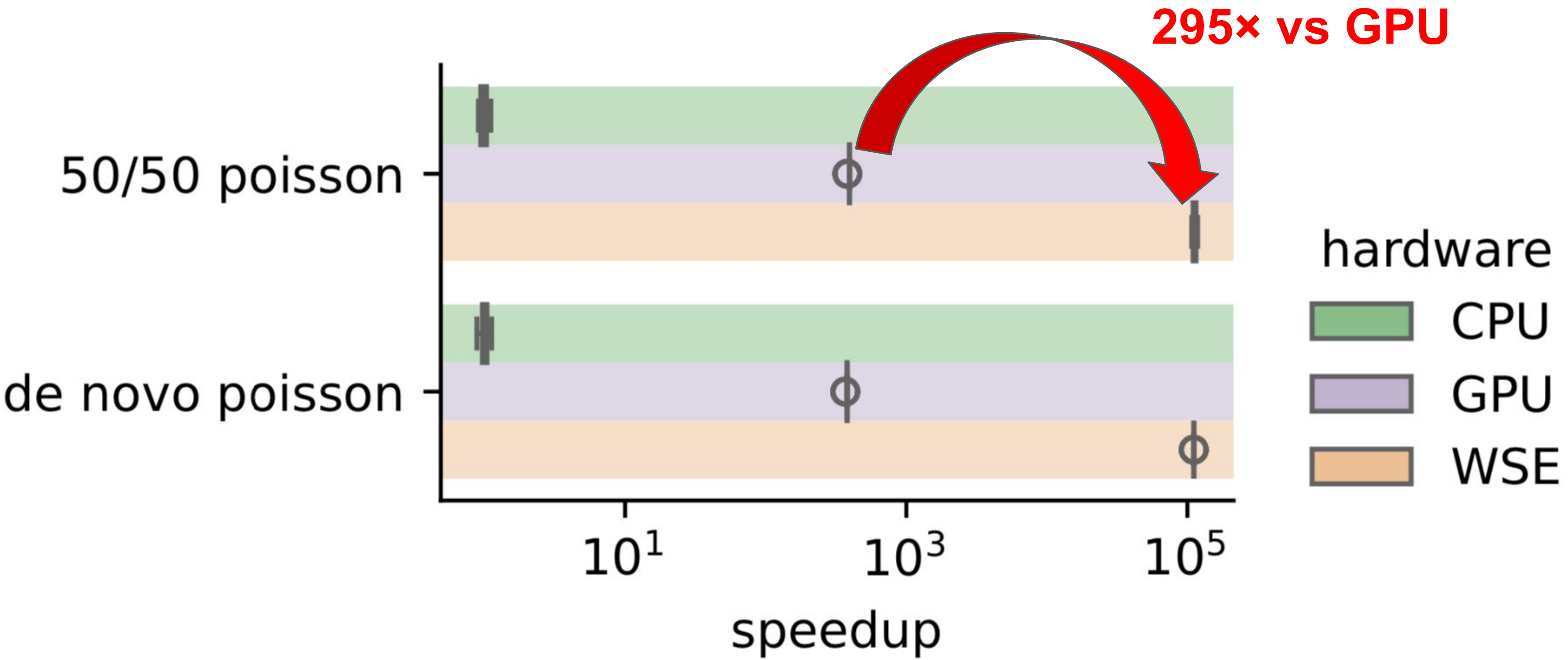
all ~8×10^{11} *(800 billion)* **agent-generations per second**

Performance



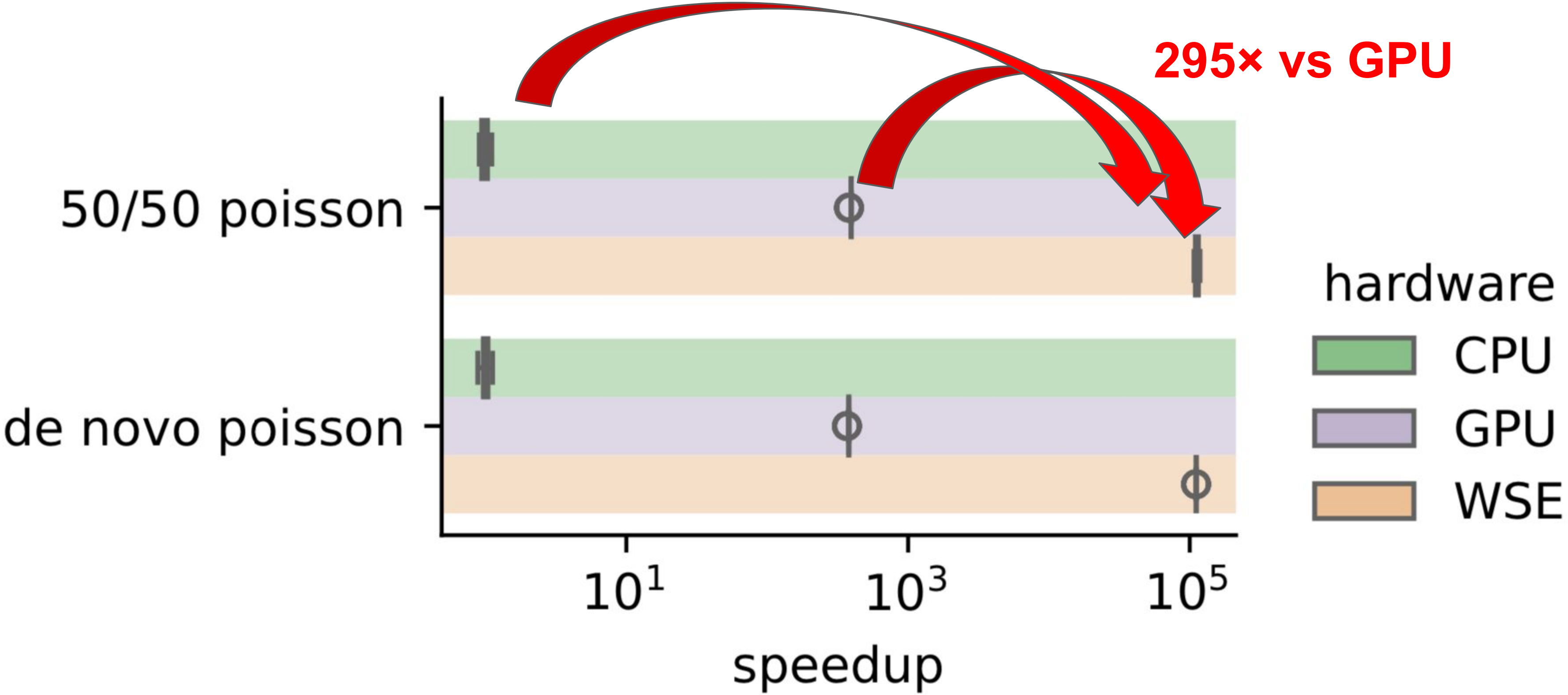
WSE ~8x10^{11} *(800 billion)* **agent-generations per second**

Performance



WSE ~8×10^{11} *(800 billion)* **agent-generations per second**

Performance



111,091× vs CPU

295× vs GPU

WSE ~8×10^{11} *(800 billion)* **agent-generations per second**

Conclusion

Summary

- Under ***unlimited*** adaptive potential, large asexual populations favor hypermutators (Raynes et al., 2018)

Summary

- Under ***unlimited*** adaptive potential, large asexual populations favor hypermutators (Raynes et al., 2018)
- Under ***restricted*** adaptive potential, normomutators regain favor in very large populations

Summary

- Under ***unlimited*** adaptive potential, large asexual populations favor hypermutators (Raynes et al., 2018)
- Under ***restricted*** adaptive potential, normomutators regain favor in very large populations
    - *This effect is amplified when hypermutators are initially rare, with strong spatial structure*

Next Steps

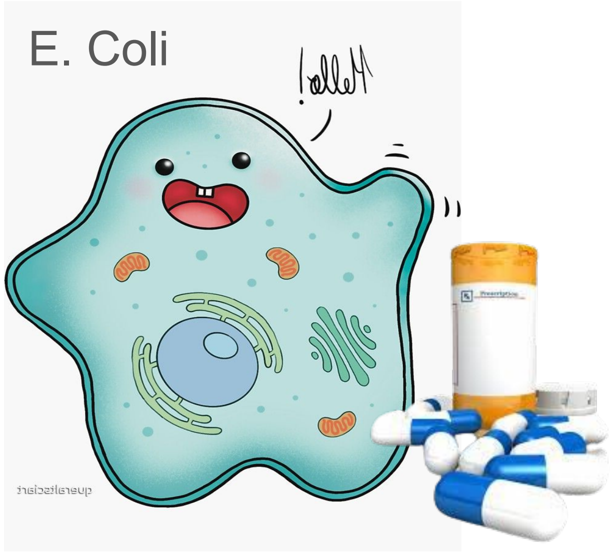
- more sophisticated fitness landscape models
- hypermutator reversion
- **connect to *in vivo* data?**

Next Steps

Martjin Callens et al., 2023

- more sophisticated fitness landscape models
- hypermutator reversion
- **connect to *in vivo* data?**

vs



osmotic pressure
more adaptive loci
→
more hypermutators

antibiotics
fewer adaptive loci
→
fewer hypermutators