POY VERSION 4

Andrés Varón

Division of Invertebrate Zoology
American Museum of Natural History
and
Computer Science Department
The Graduate Center
City University of New York

GOALS OF THE PROJECT

- Multiple phylogenetic inference criteria
- Support from pre-aligned sequences, to complete genomes, developmental sequences, and morphology
- Good performance
- Analytic, educational, and research tool
- High quality control

THIS TALK

- Phylogenetic analysis features
- Performance
- Flexibility
- Quality control

PHYLOGENETIC ANALYSIS FEATURES

TRANSFORMATIONS SUPPORTED

- Substitutions
- Insertions and Deletions
- Inversions
- Translocations
- Horizontal Gene Transfer
- Other transformations

GOALS

	Static	Dynamic Homologies	
	Matrix	Unaligned	Rearrangements
Parsimony			
Likelihood			
Bayesian			

POY VERSION 3

	Static	Dynamic Homologies	
	Matrix	Unaligned	Rearrangements
Parsimony			
Likelihood			
Bayesian			

POY VERSION 4

	Static	Dynamic Homologies	
	Matrix	Unaligned	Rearrangements
Parsimony			
Likelihood			
Bayesian			

ALGORITHMS

- **Random Addition Sequence**
- SPR
- **TBR**
- Sectorial Search
- Tree Fusing
- Ratchet
- Perturbation
- Simulated Annealing
- Tree Drifting
- Branch and Bound
- Multiple new heuristics
- **Direct Optimization**
- Affine-DO
- Fixed States
- Iterative improvement
- Exhaustive (Affine-)DO
- Local search for GTAP with rearrangements

POY VERSION 4

	Static	Dynamic Homologies	
	Matrix	Unaligned	Rearrangements
Parsimony			
Likelihood			
Bayesian			

PERFORMANCE

COMMON COMMENTS

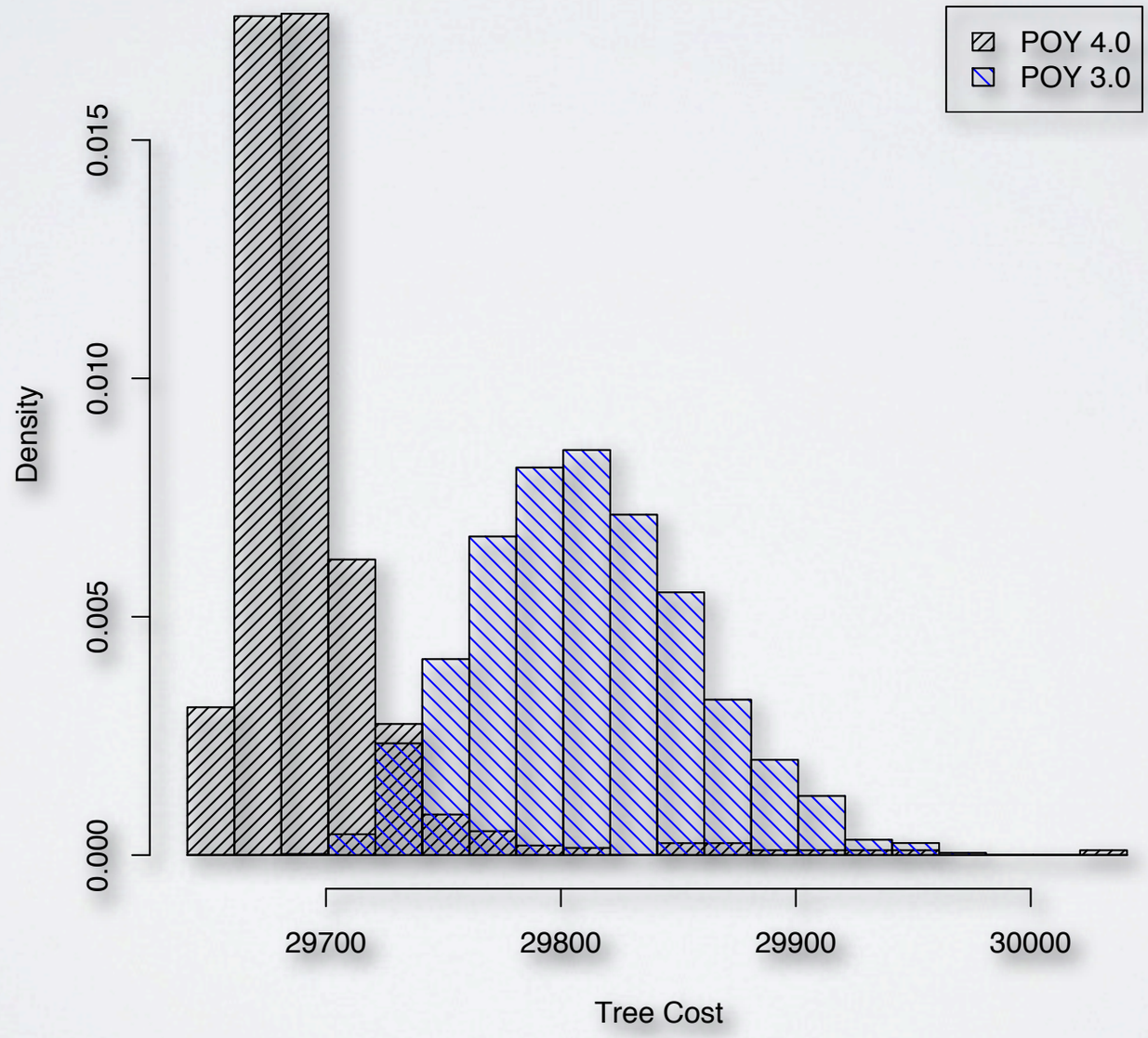
- POY is slow (needs a cluster)
- POY's trees are very inaccurate
- POY doesn't scale (only small data sets really)

VERSION COMPARISON

Version 3 with Version 4
100 terminals, 8 genes, 35 morphological characters
(subset of Faivovich *et al*., 2005)
1000 iterations Random Addition Sequence followed by TBR

SPEED COMPARED WITH POY 3

Random Addition Sequence followed by TBR



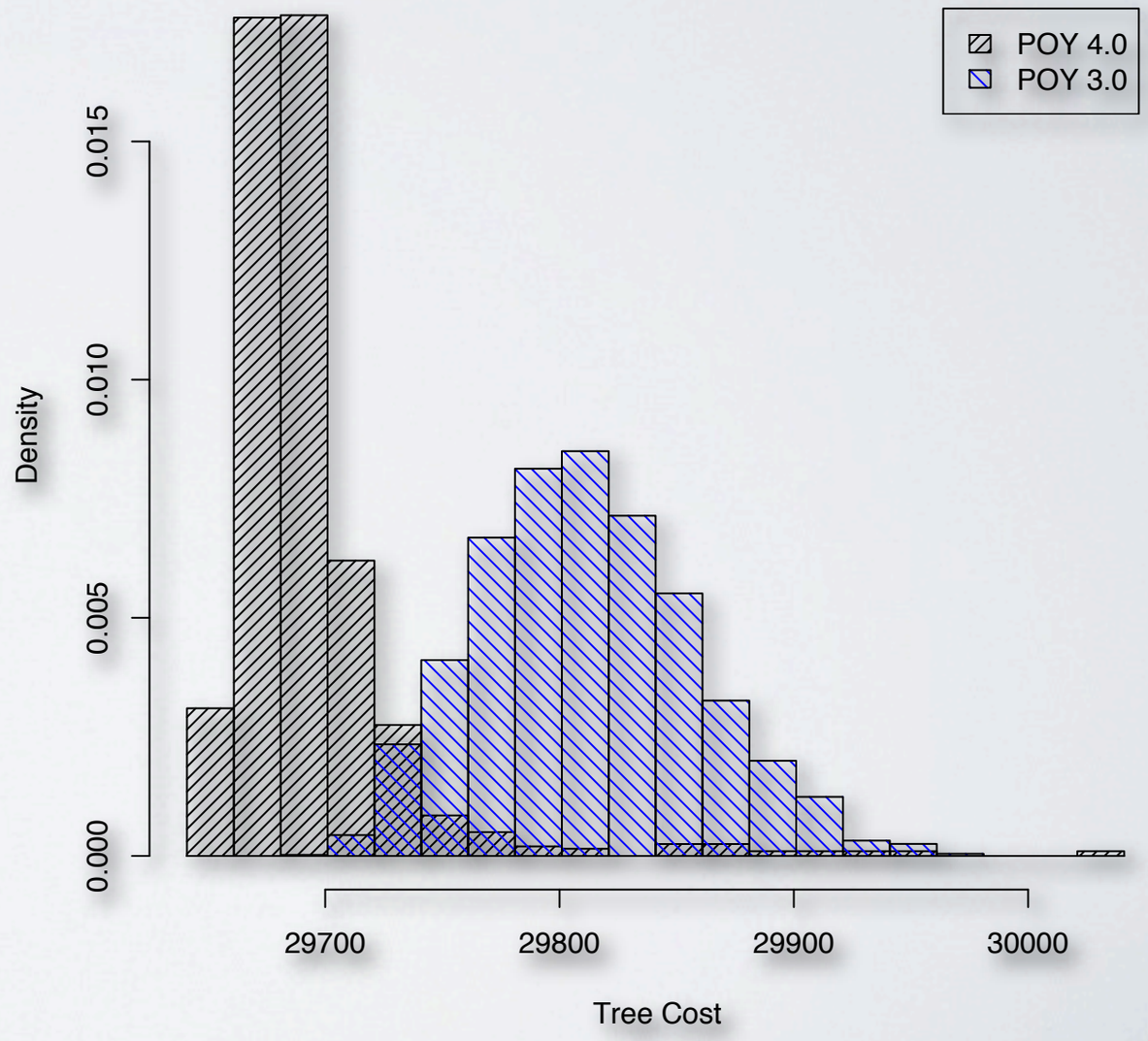
SPEED COMPARED WITH POY 3

Random Addition Sequence followed by TBR

POY 4 + Dual core workstation

=

POY 3 + 5000 core cluster



COMMON COMMENTS

- POY is slow (needs a cluster)
- POY's trees are very inaccurate (using affine indels)
- POY doesn't scale (only small data sets really)

COMMON COMMENTS

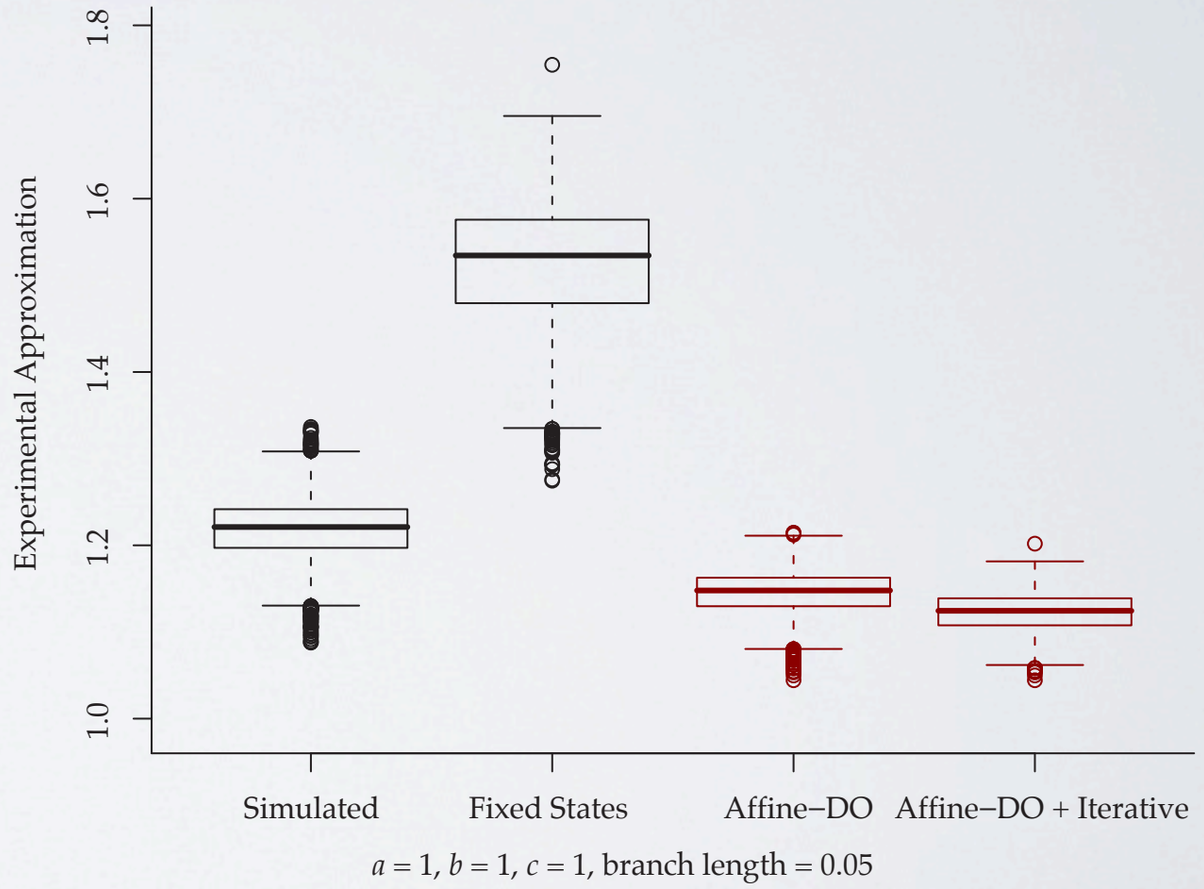
- ~~POY 3 is slow (needs a cluster)~~ POY 4 doesn't need one
- POY's trees are very inaccurate (using affine indels)
- POY doesn't scale (only small data sets really)

COMPARED WITH PRE-ALIGNED SEQUENCE

- Ogden and Rosenberg, 2007 (POY 3)
  - POY = 10 RAS + TBR with non-affine gap costs
  - Simulate with affine gaps
- None of the pre-aligned sequence methods support affine gaps as transformation events

IMPROVED AFFINE GAP SUPPORT IN POY 4

- Only program of this performance and scalability for (affine) tree alignment



COMPARISONS WITH PRE-ALIGNED SEQUENCES

- Lehtonen, 2008
  - POY's inference is better even using non-affine cost with a better search (using POY 4).
- Wheeler, 2009
  - POY's trees are much shorter.
- Liu, *et al.*, 2009
  - POY's phylogenies with unaligned sequences are very competitive.

COMPARED WITH PRE-ALIGNED SEQUENCE

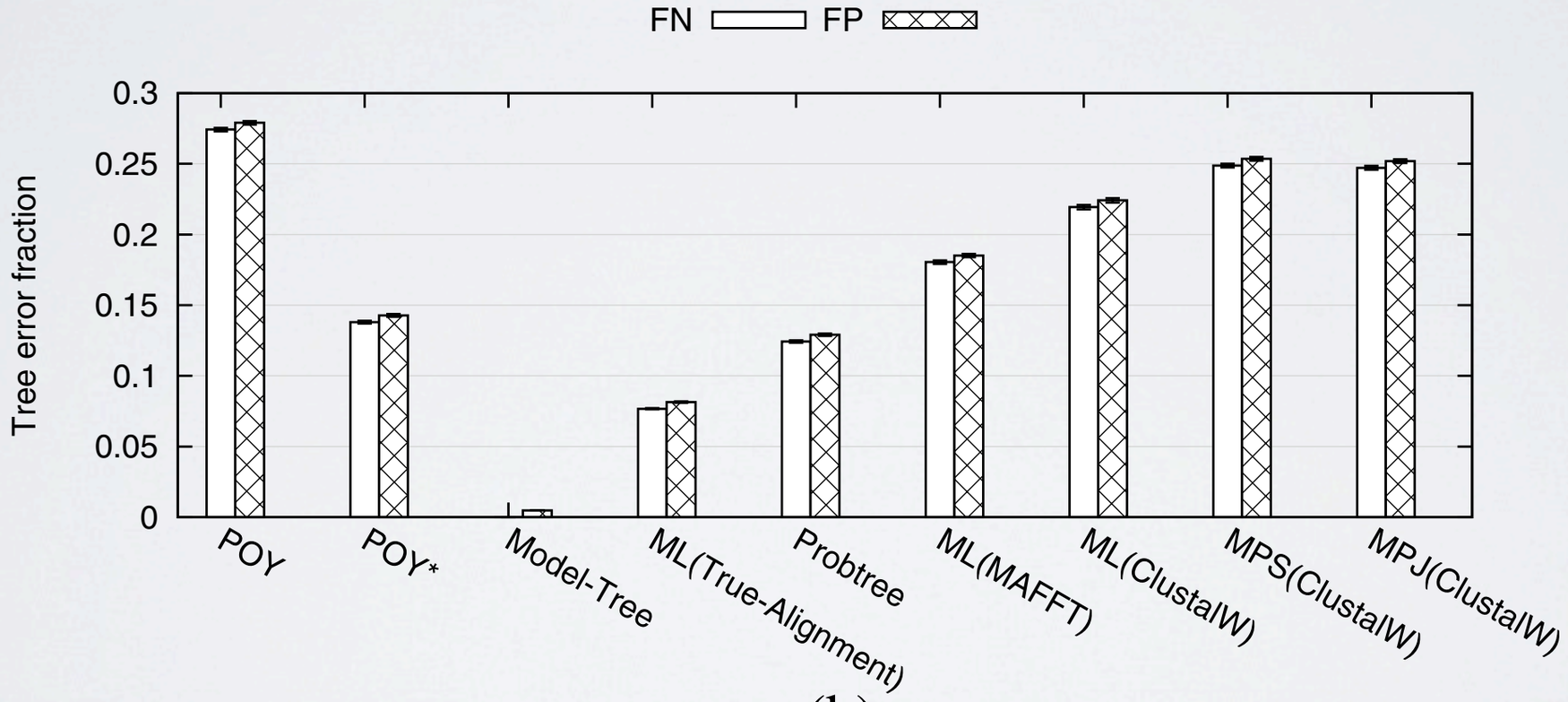
ML = RAxML with GTRMIX

PS (X) = POY score for a tree generated by method X

POY = 1 Random Addition Sequence followed by TBR

POY* = Probtree + TBR in POY

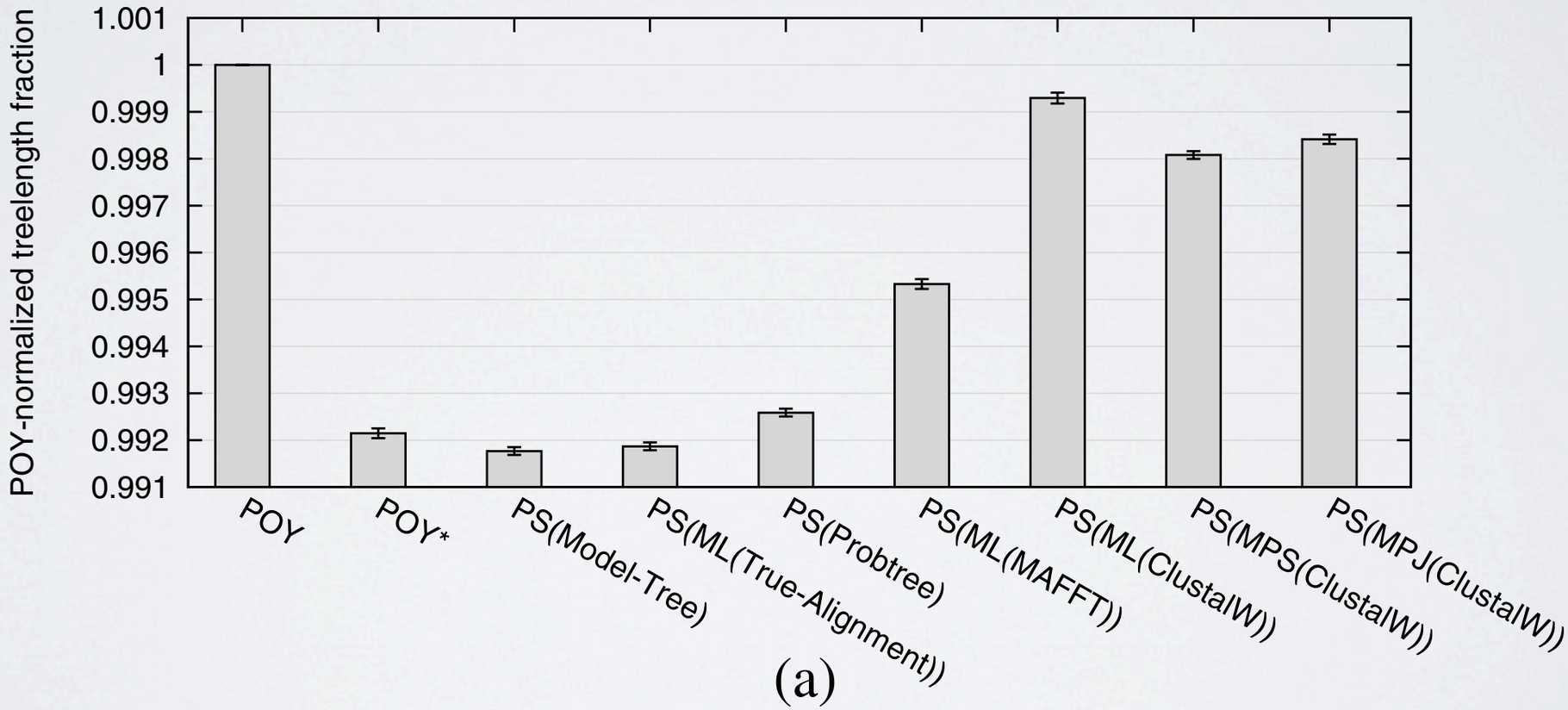
COMPARED WITH PRE-ALIGNED SEQUENCE



(b)

Liu, *et al.*, 2009

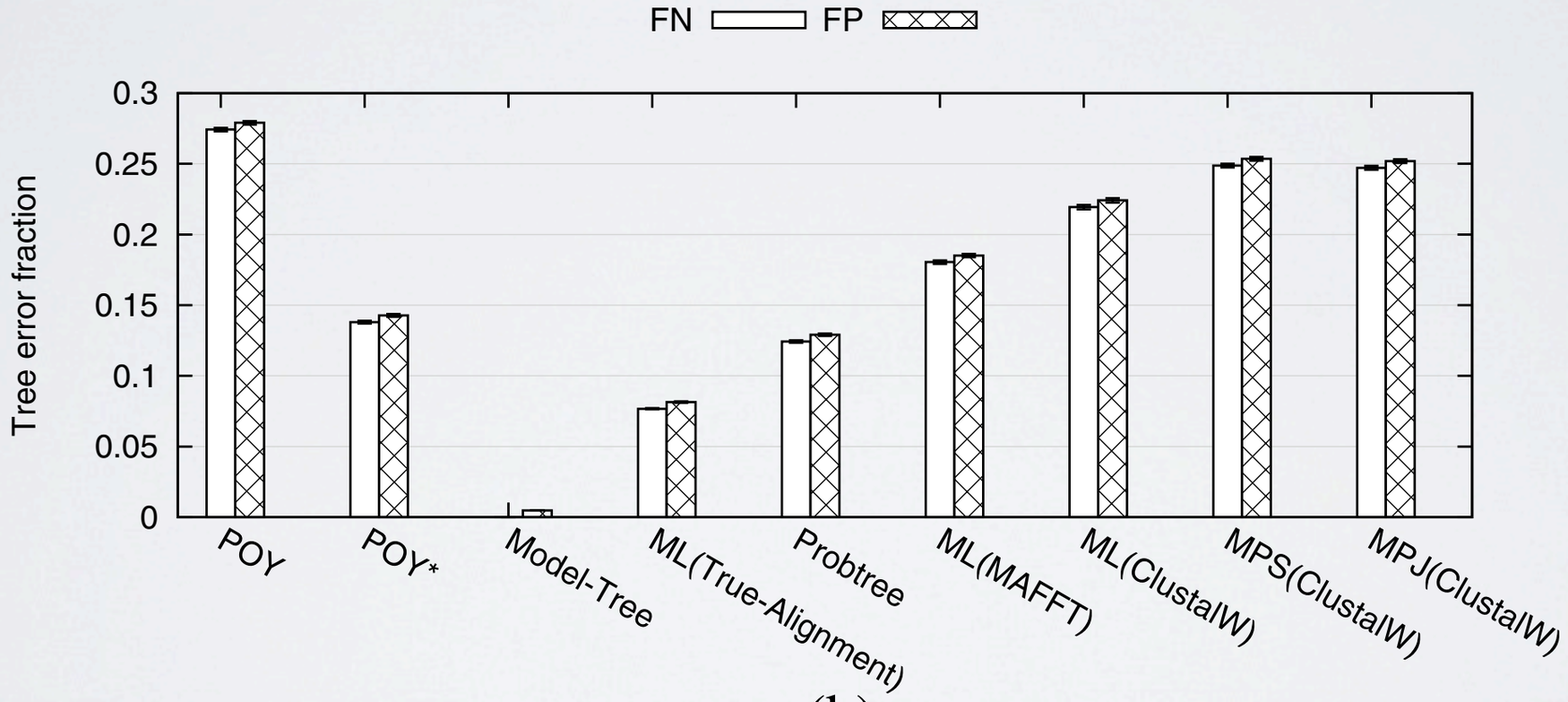
COMPARED WITH PRE-ALIGNED SEQUENCE



(a)

Liu, *et al.*, 2009

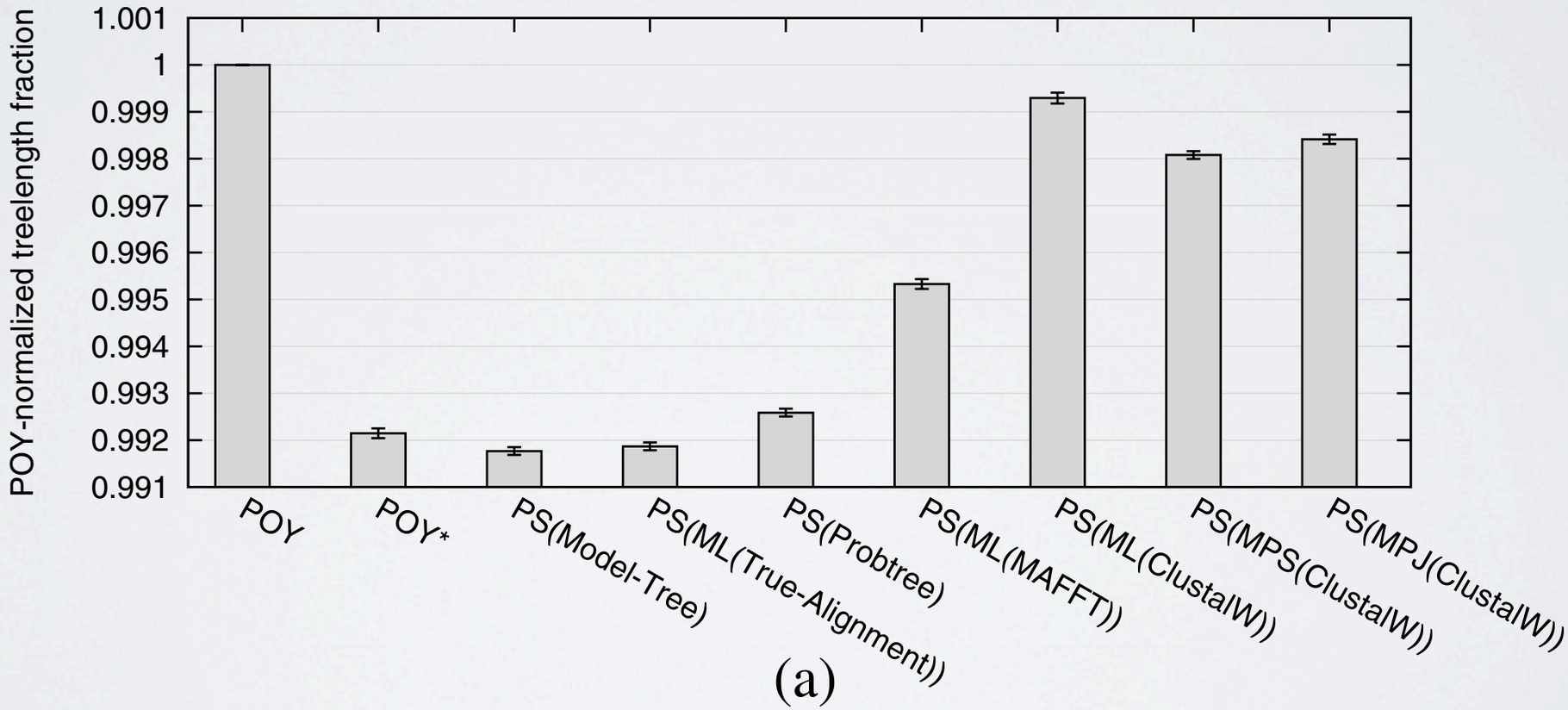
COMPARED WITH PRE-ALIGNED SEQUENCE



(b)

Liu, *et al.*, 2009

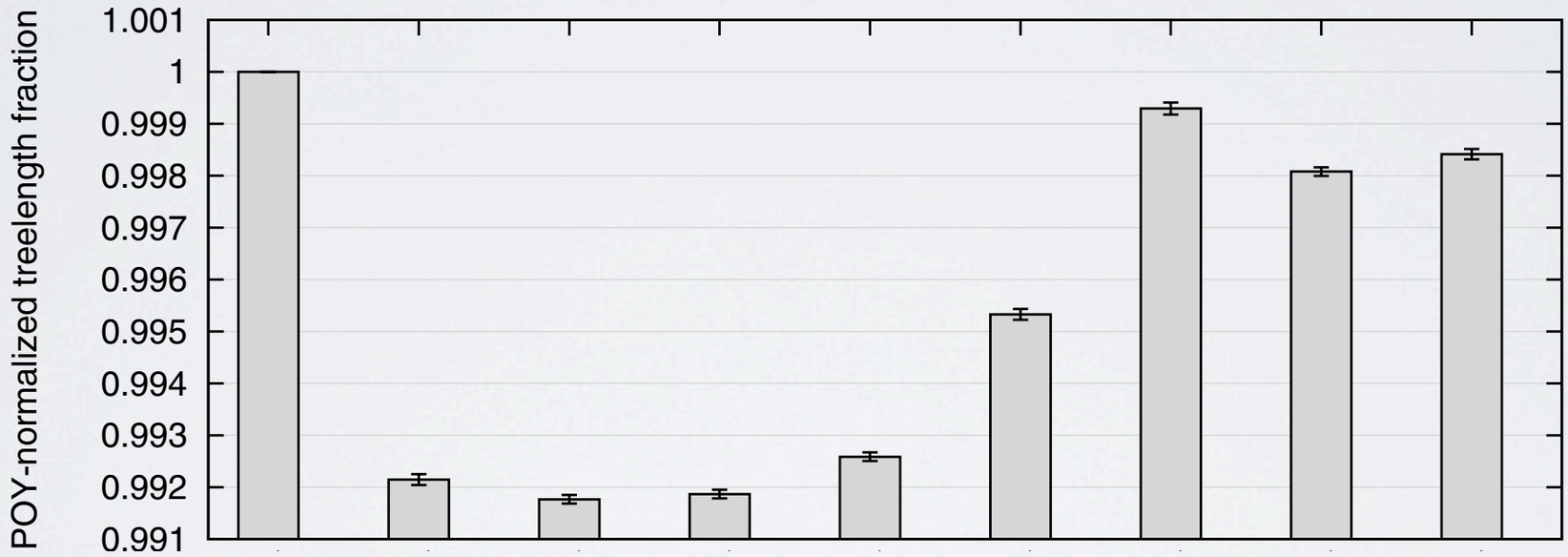
COMPARED WITH PRE-ALIGNED SEQUENCE



(a)

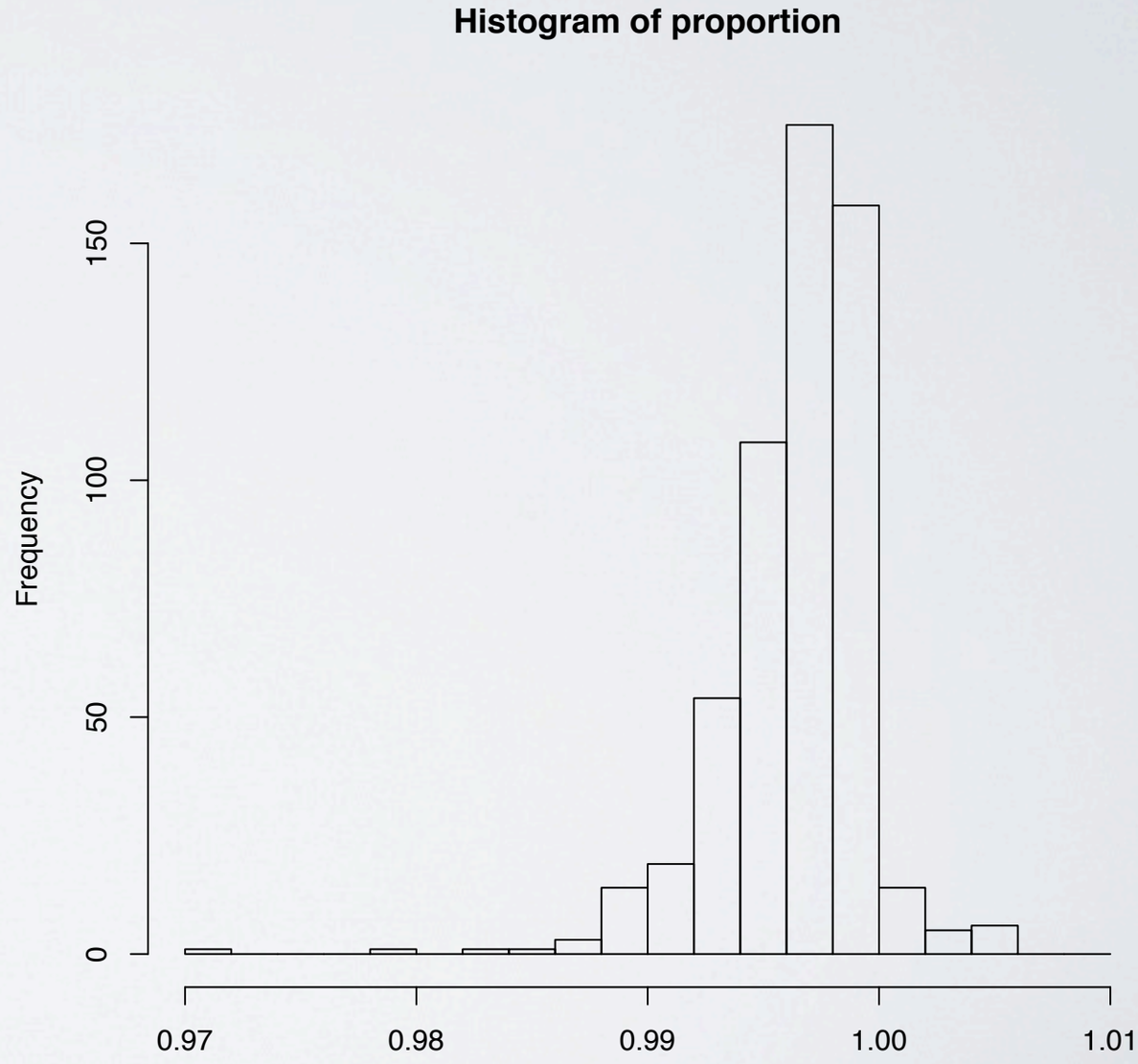
Liu, *et al.*, 2009

COMPARED WITH PRE-ALIGNED SEQUENCE



DOES POY DO WELL?

- Better heuristic in POY
- Time limit of 2 hours
- POY produces shorter trees than POY*



COMMON COMMENTS

- ~~POY 3 is slow (needs a cluster)~~ POY 4 doesn't need one
- POY's trees are very inaccurate (using affine indels)
- POY doesn't scale (only small data sets really)

COMMON COMMENTS

- ~~POY 3 is slow (needs a cluster)~~ POY 4 doesn't need one
- POY's trees are ~~very inaccurate~~ better
- POY doesn't scale (only small data sets really)

SCALABILITY

SCALABILITY METHODS

- New algorithms
- Functional programming and data structures
- Script analysis and optimization

FUNCTIONAL PROGRAMMING

- No global variables (well .... there are two counters).
- No side effects (interfaces are purely functional).

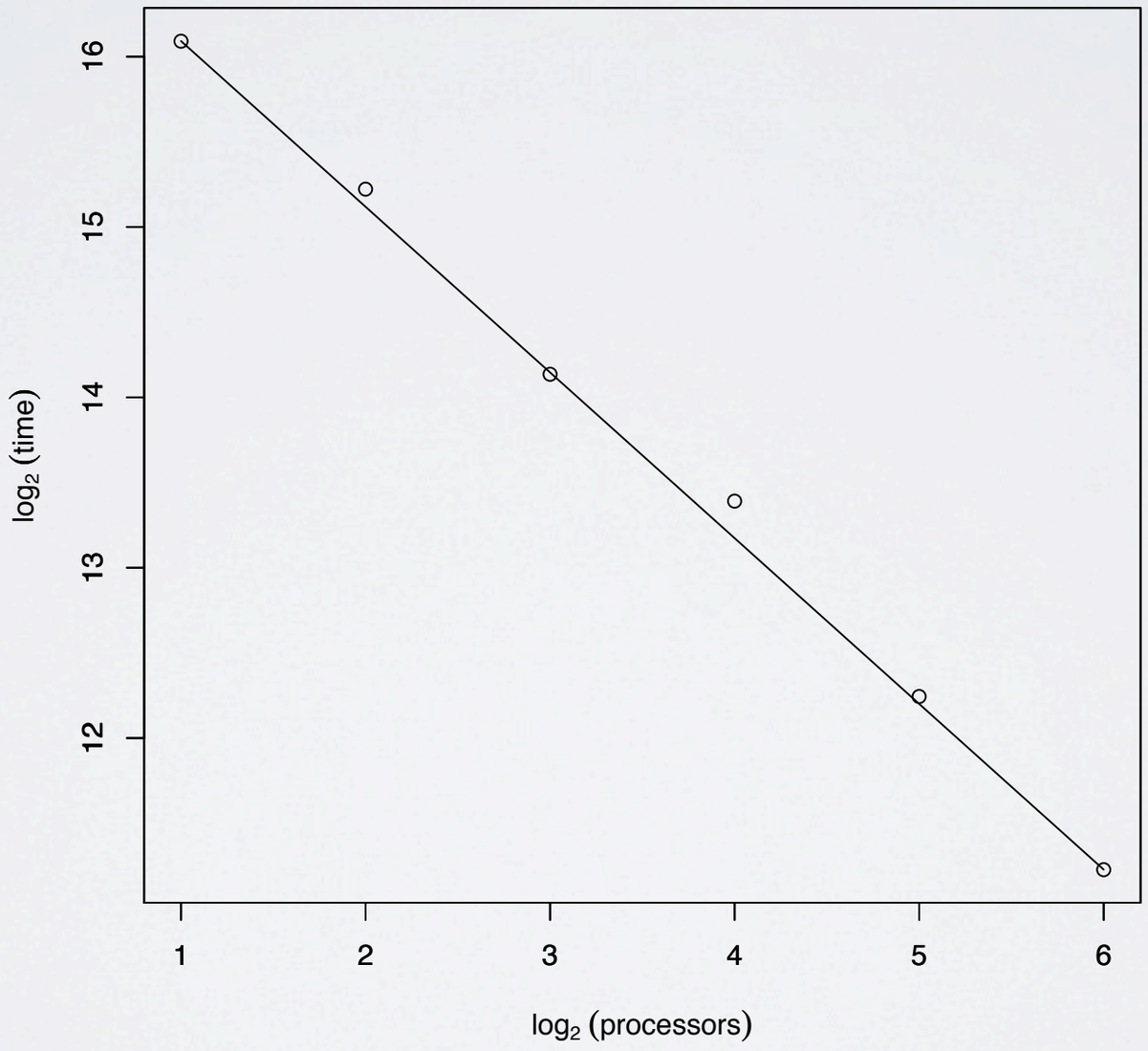
SCRIPT ANALYSIS AND OPTIMIZATION

```
read ("file")              (* Non-Composable *)
build (1000)               (* Parallelizable *)
swap ()                    (* Parallelizable *)
redraw ()                  (* Linnearizable  *)
select ()                  (* Composable     *)
report (graphtrees)        (* Non-Composable *)
quit ()                    (* Non-Composable *)
```

SCRIPT ANALYSIS AND OPTIMIZATION

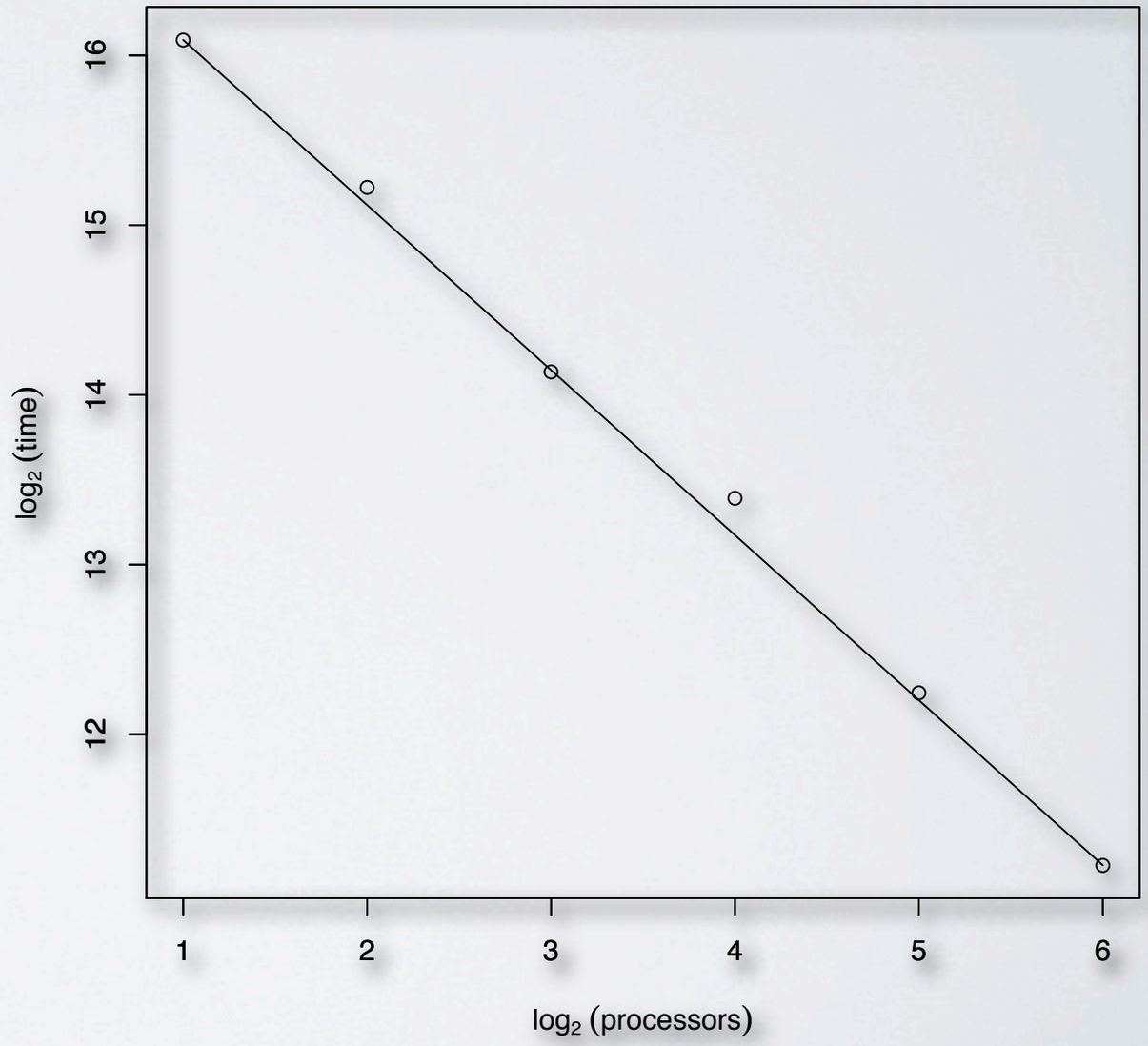
```
beginning of the program
read an input file
I will calculate the following in separate processors (if available)
processor group 1:
  in parallel:
    build some trees from scratch
    swap the trees in memory
    while keeping the following invariant:
      eliminate repeated trees
      select the optimal trees
  eliminate repeated trees
  select the optimal trees
processor group 2:
  redraw the screen
  output the trees in memory
  close POY
```

SCRIPT ANALYSIS AND OPTIMIZATION



SCALABILITY EXAMPLE

- Linear scalability + limited memory consumption



LARGEST ANALYSES
(TO MY KNOWLEDGE)

- In terminals: ~1.700 terminals, 4 genes (~ 4.000 bp)

- Simulations of 1.000 sequences in a modern workstation within 72 hours

- In genome length: >800.000 bp and 342 genes for 6 terminals

- Linear scalability in parallel execution

COMMON COMMENTS

- ~~POY 3 is slow (needs a cluster)~~ POY 4 doesn't need one
- POY's trees are ~~very inaccurate~~ better
- POY doesn't scale (only small data sets really)

COMMON COMMENTS

- ~~POY 3 is slow (needs a cluster)~~ POY 4 doesn't need one
- POY's trees are ~~very inaccurate~~ better
- ~~POY doesn't scale (only small data sets really)~~

POY 4 ADVANTAGES

- Shorter time
- Better tree costs
- Indels consistently treated within the optimality criterion

WARNING!

- The simulations are easy to attack
- The results depend on the model used
- The data are ideal, not real (e.g. patterns of missing fragments)

FLEXIBILITY

FLEXIBILITY

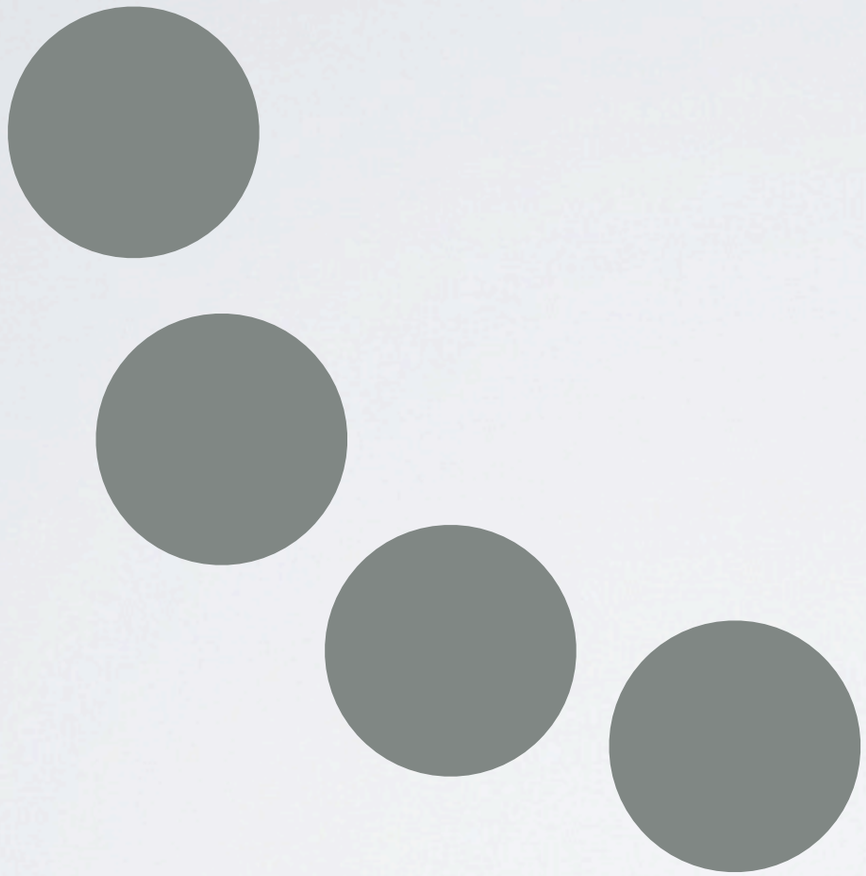
- Open source
- OCaml and C
- Plugin architecture to add functions
- New character types can be easily added
- Extensions to the Objective CAML language to inject POY scripts
- Extensive documentation
- Many file formats supported

QUALITY CONTROL

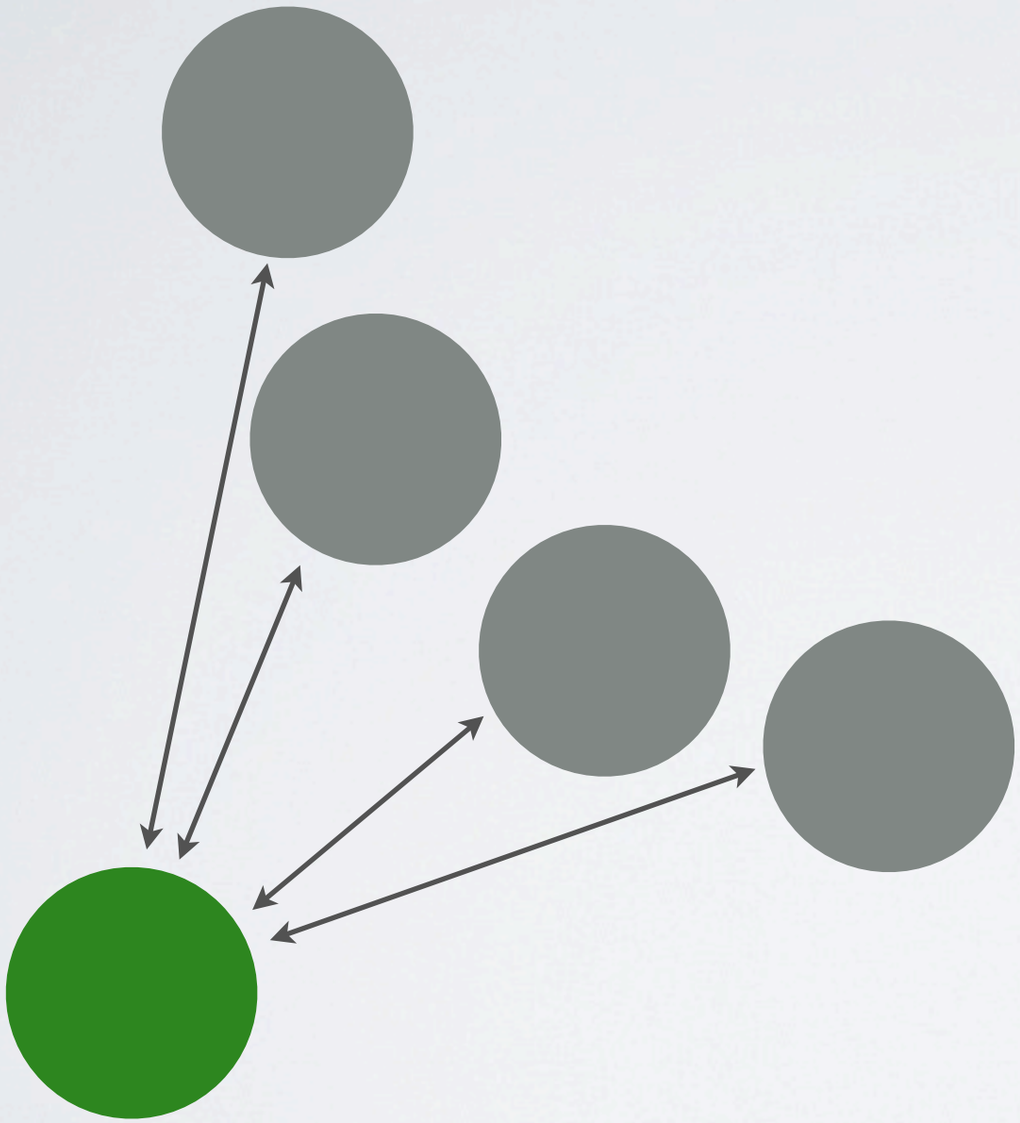
QUALITY CONTROL

- OCaml helps *a lot*
- Release early, release often
- Provide very active support to users
- Distributed unit tests in multiple architectures with distributed version control

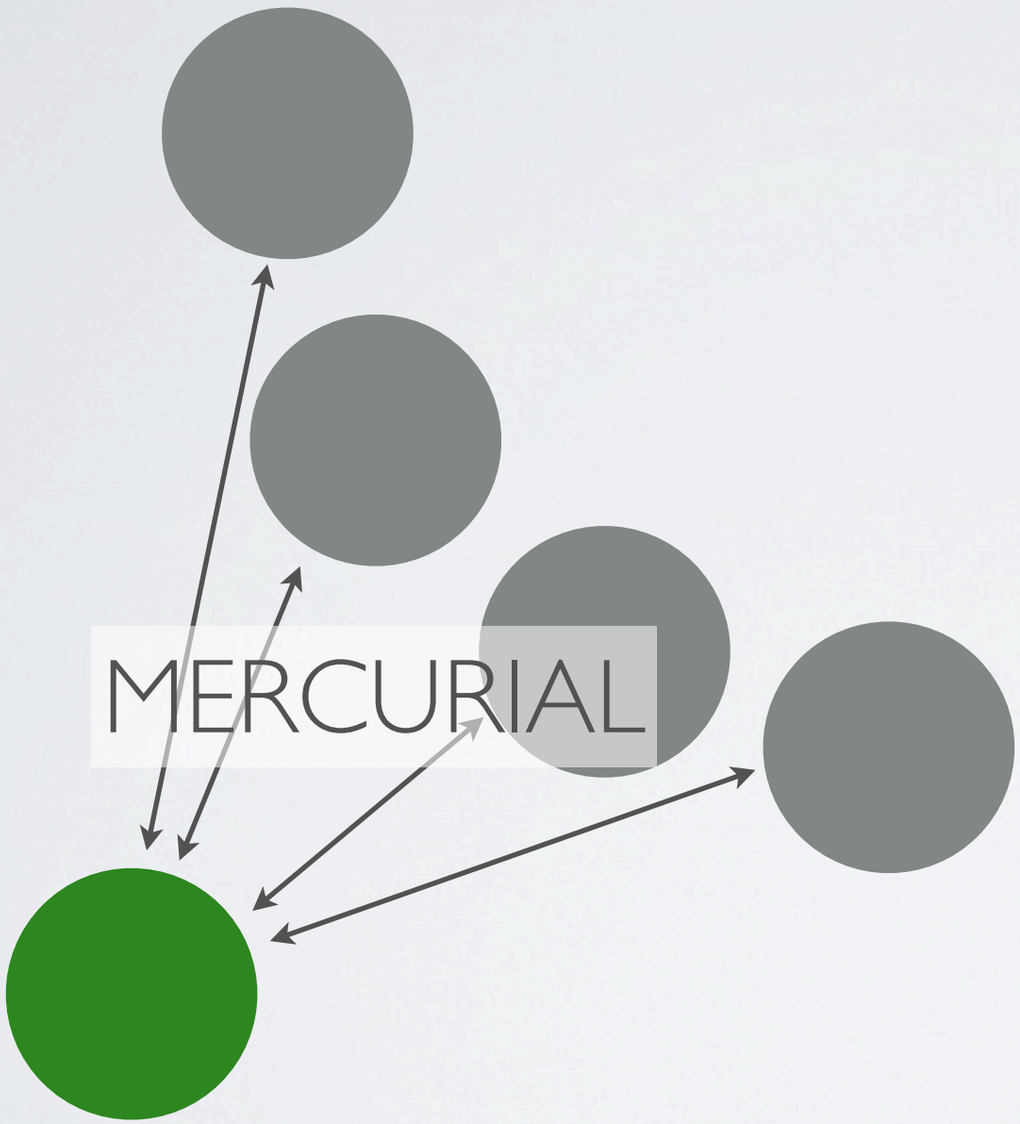
QUALITY CONTROL



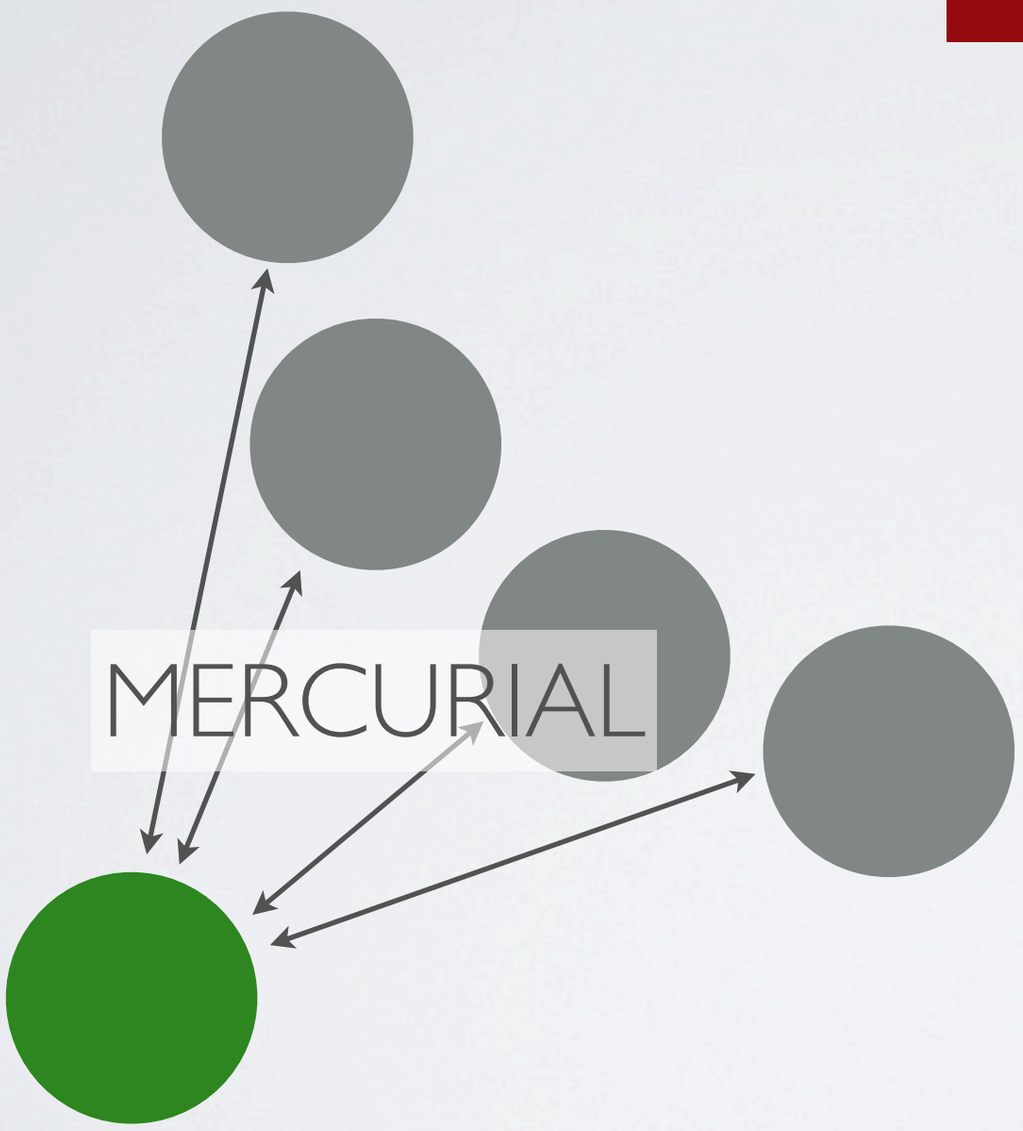
QUALITY CONTROL

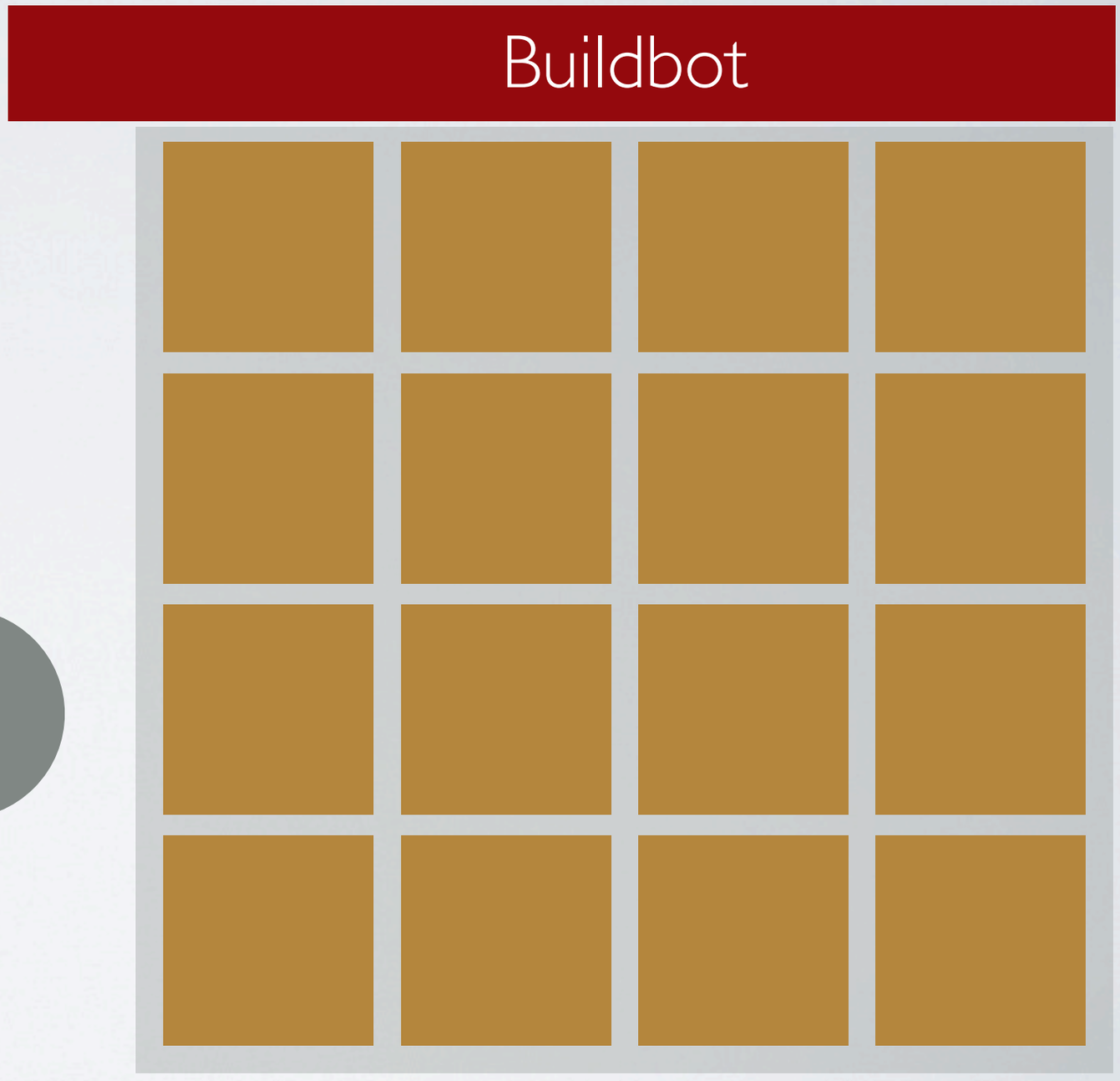


QUALITY CONTROL

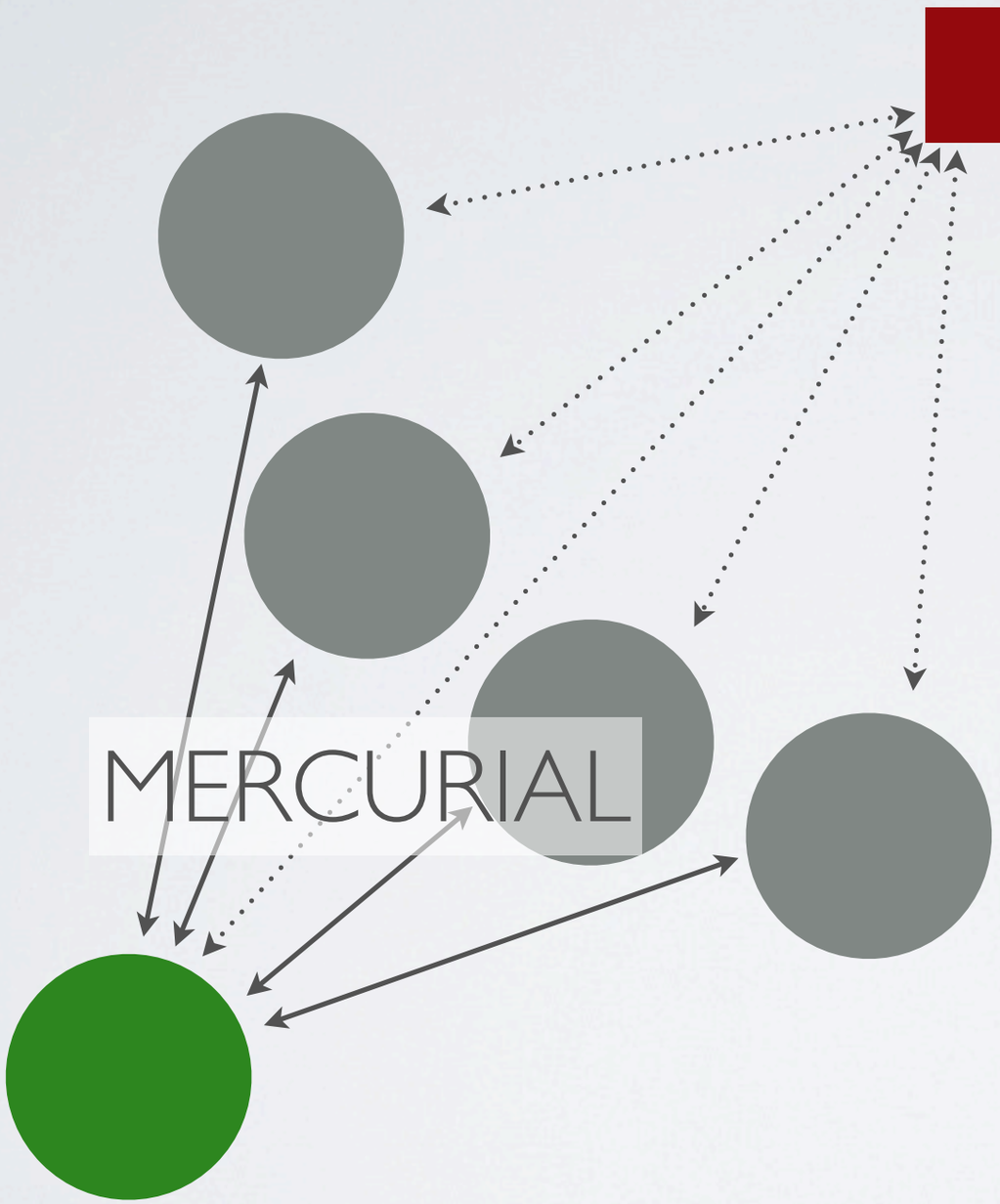


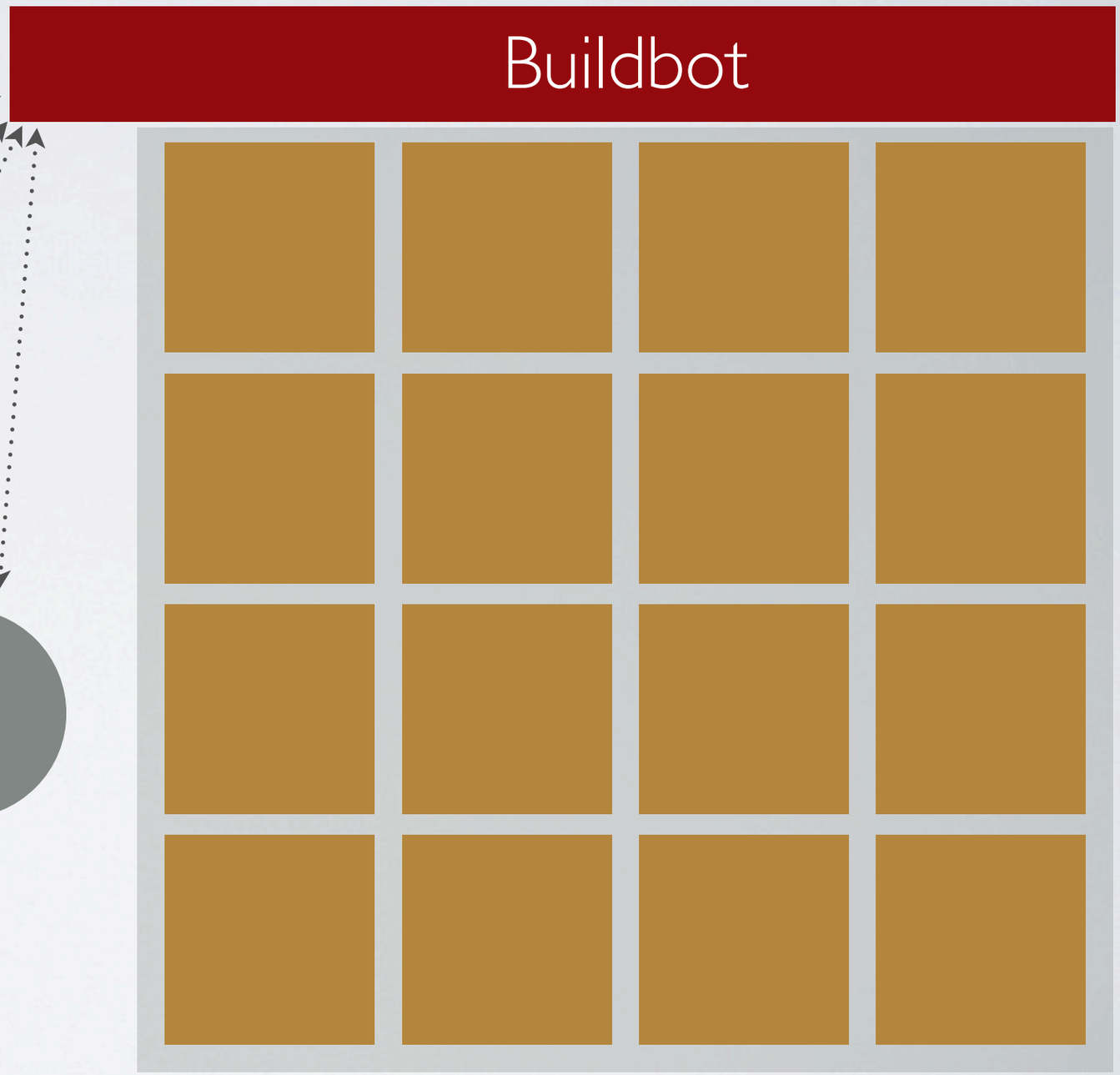
QUALITY CONTROL



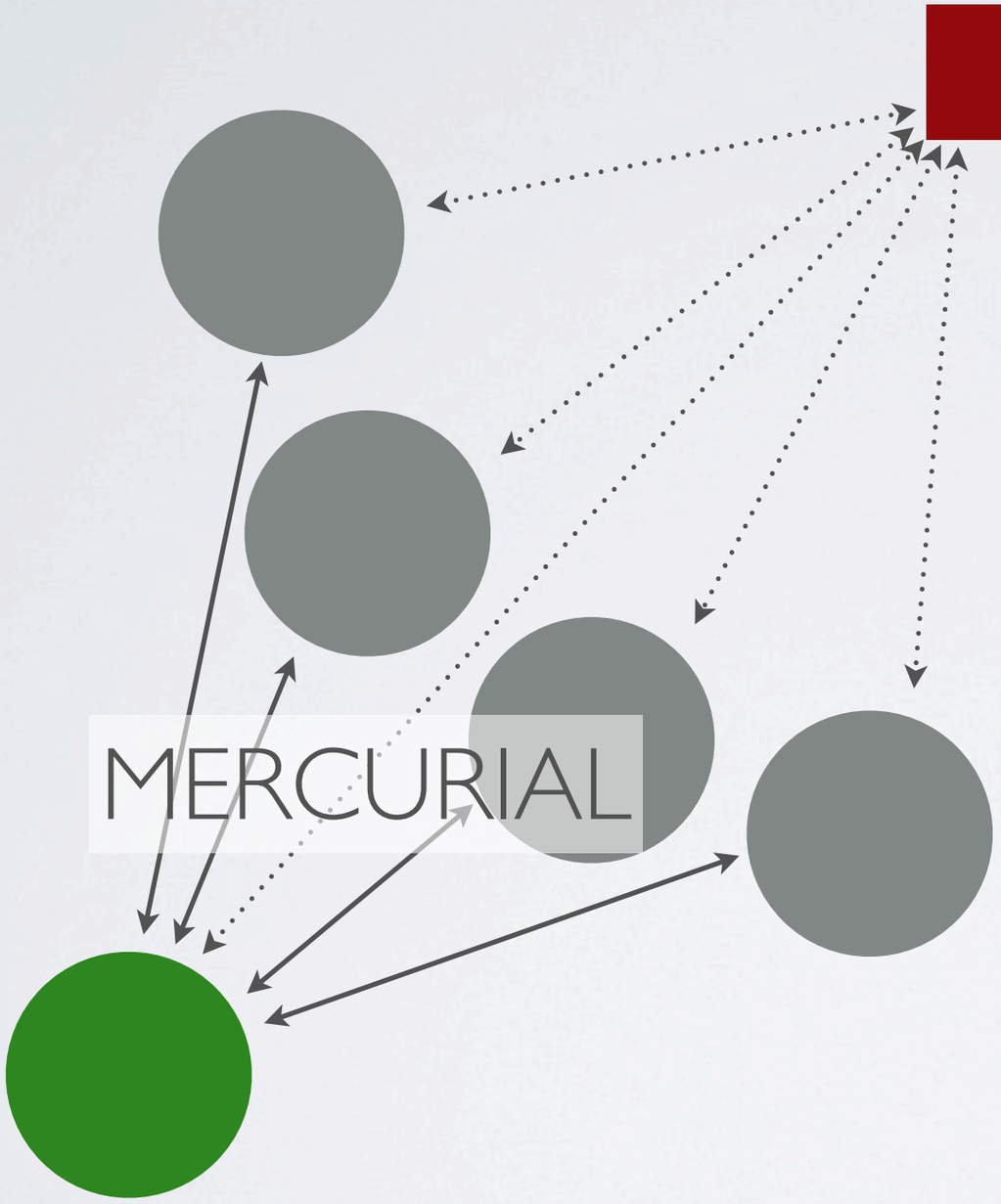


QUALITY CONTROL





QUALITY CONTROL

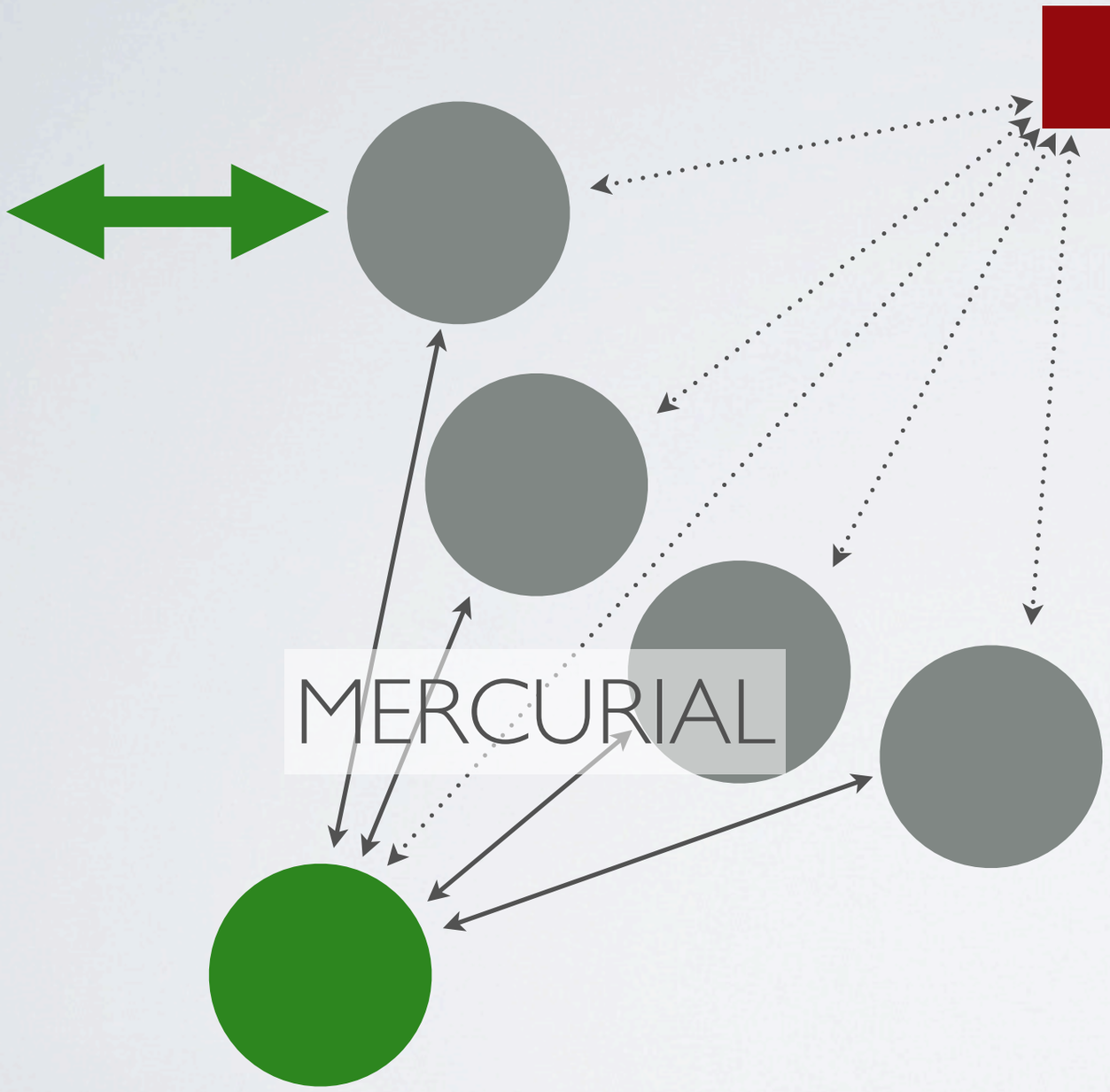


Buildbot

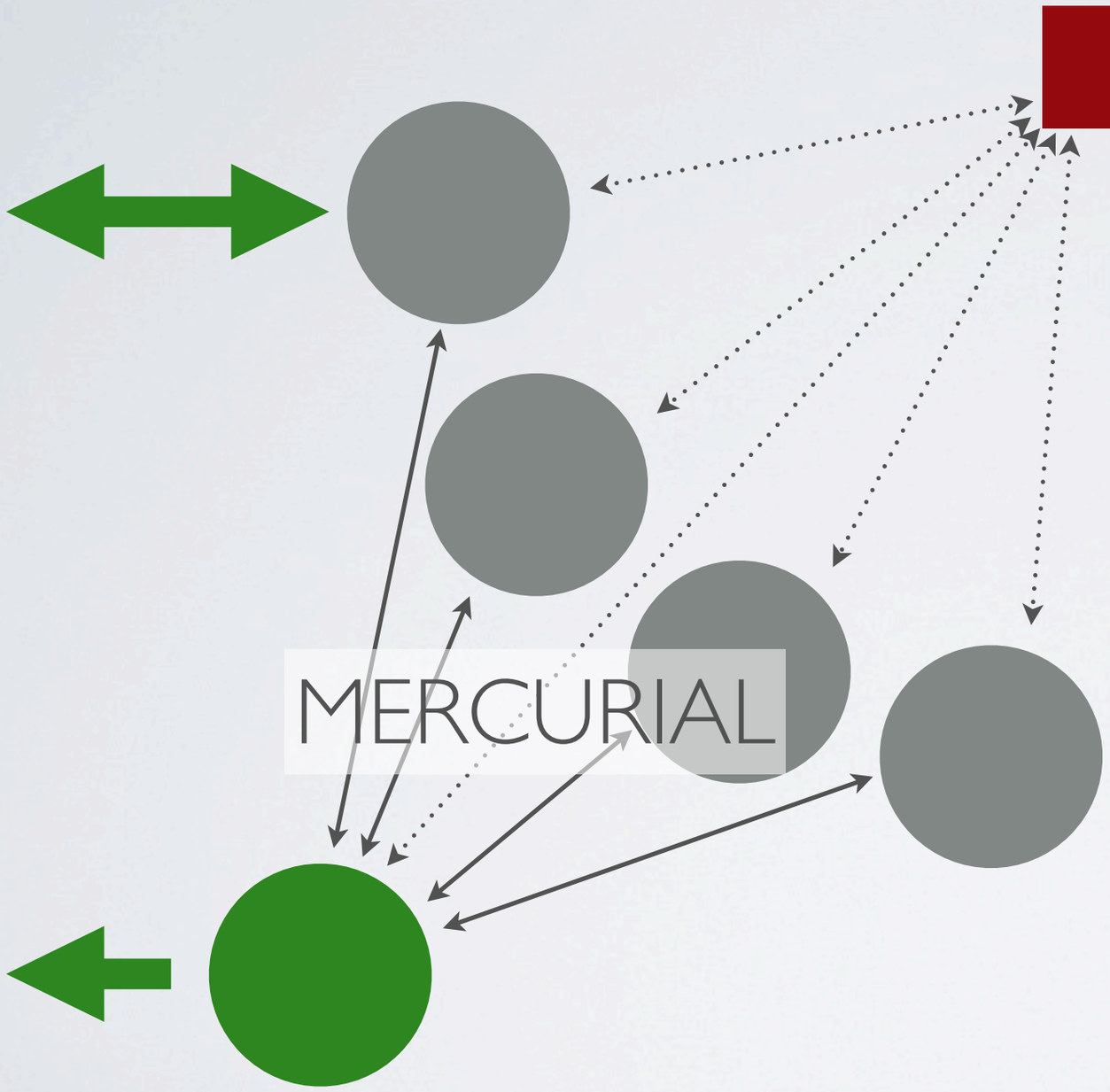
ON COMMIT
NIGHTLY
WEEKELY

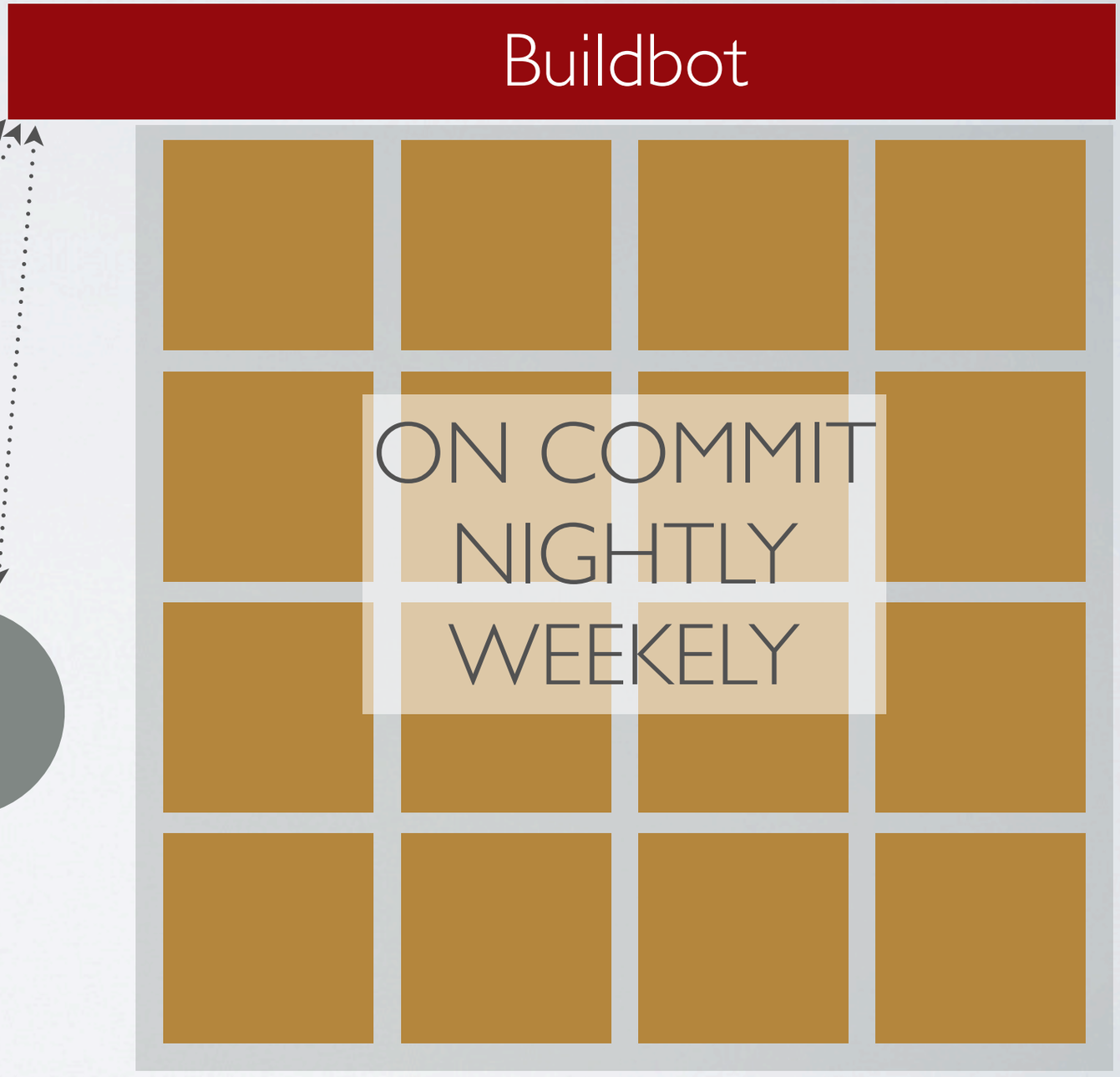
Test Farm

QUALITY CONTROL



QUALITY CONTROL





SUMMARY

- Phylogenetic analysis features
- Performance
- Flexibility
- Quality control

ACKNOWLEDGMENTS

- Le Sy Vinh
- Nicholas Lucaroni
- Ilya Bomash
- Lin Hong
- Illya Temkin
- Megan Cevasco
- Julian Faivovich
- Kurt M. Pickett
- Taran Grant
- William Smith
- Luisa Paola Pedraza
- Dan Janies
- All the users that have tested and provided feedback

ACKNOWLEDGMENTS

- DoD - DARPA
- NSF - CIPRES
- American Museum of Natural History

POY 4

Website:

http://research.amnh.org/scicomp/projects/poy.php

Mailing list:

http://groups.google.com/group/poy4/

Source code and bug reports:

http://code.google.com/p/poy4/

TRANSFORMATIONS

CCTCCAATGATACGTTGAAAGGCGTTTATCGT

TRANSFORMATIONS

CCTCCAATGATACATTGAAAGGCGTTTATCGT

TRANSFORMATIONS

CCTCCA**C**TGATAC**A**TTGAAAGGCGTTTATCGT

TRANSFORMATIONS

CCTCCA**C**TGAAC**A**TTGAAAGGCGTTTATCGT

TRANSFORMATIONS

CCTCCA**C**TGAAC**A**TTGAA**T**GGCGTTTATCGT

TRANSFORMATIONS

CCTCCA**C**TGAAC**A**TTGA**C**A**T**GGCGTTTATCGT

TRANSFORMATIONS

CCTCCA**C**TGAAC**A**TTGA**C**A**T**GGTTATCGT

TRANSFORMATIONS

CCT**A**CA**C**TGAAC**A**TTGA**C**A**T**GGTTATCGT

TRANSFORMATIONS

CCTCCAATGATACGTTGAAAGGCGTTTATCGT

CCT**A**CA**C**TGAAC**A**TTGA**C**A**T**GGTTATCGT

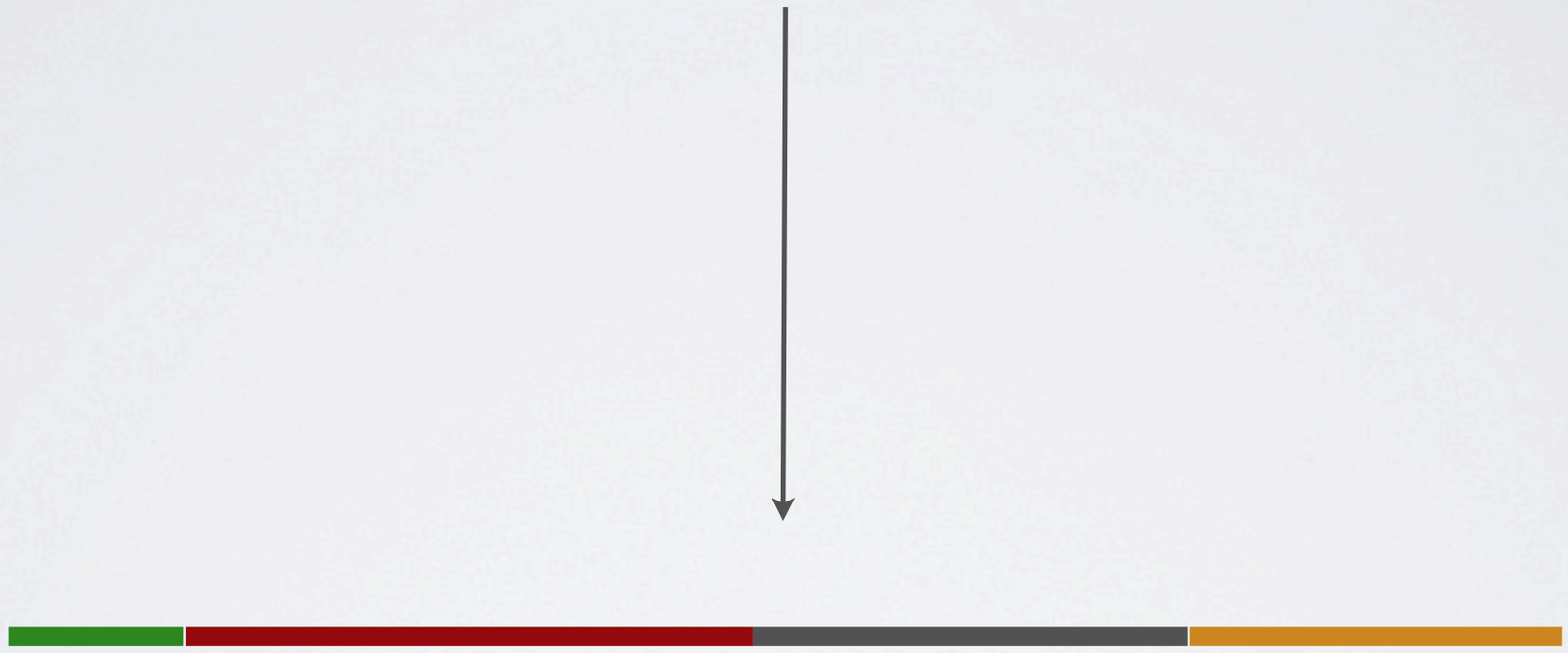
TRANSFORMATIONS

CCTTGGTTCTCTTTACTGAGTGTCTTGGGCGACCGGCACGTTTACTTTGAAAAAATT

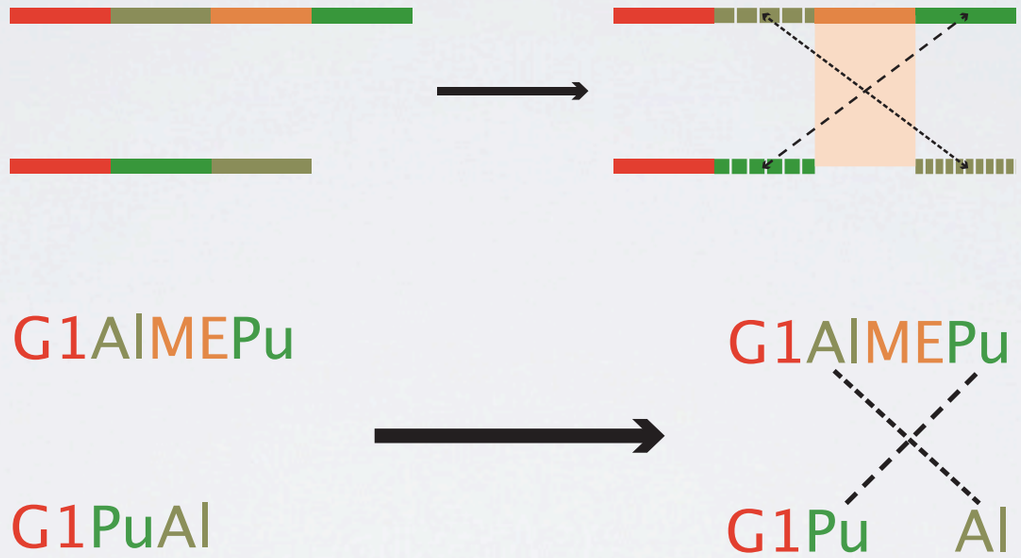
TRANSFORMATIONS

TRANSFORMATIONS

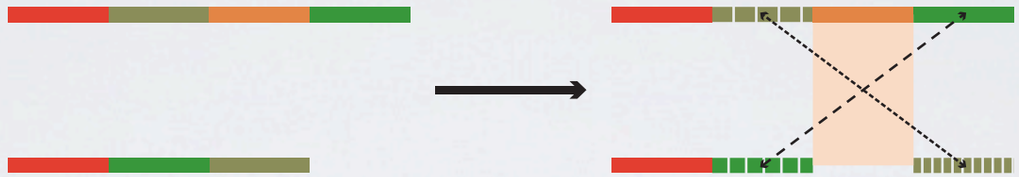
TRANSFORMATIONS



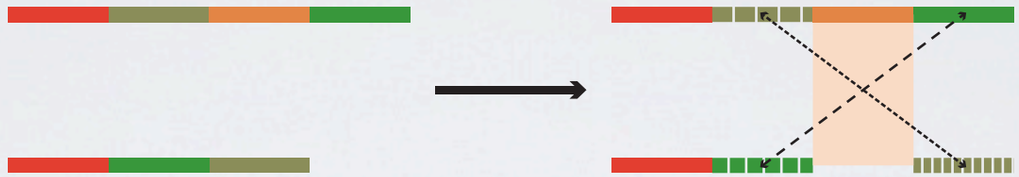
WHAT KIND OF ANALYSES POY SUPPORT?

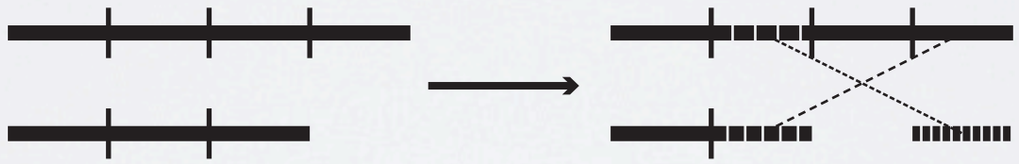


REARRANGEMENTS

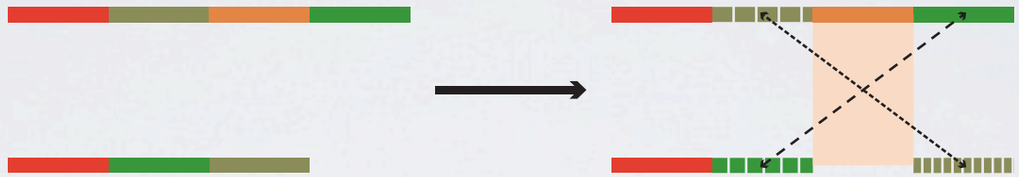


REARRANGEMENTS

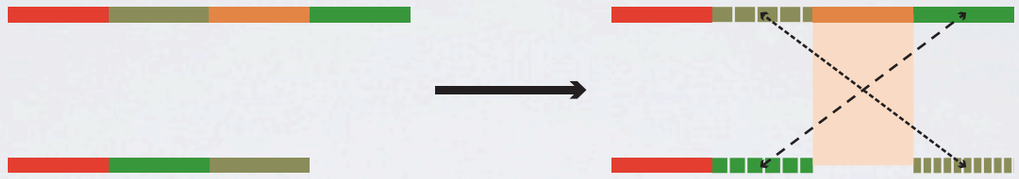




REARRANGEMENTS



REARRANGEMENTS



Breakpoint
Inversion
Double Cut and Join