Contained Chaos: Ensemble Consistency Testing for the Community Earth System Model

Dorit Hammerling

Department of Applied Mathematics and Statistics, Colorado School of Mines

with Allison Baker (Applications Scalability and Performance Group, National Center for Atmospheric Research), Daniel Milroy, Stephen Molinari, Galen Vincent, Teo Price-Broncucia and many others!

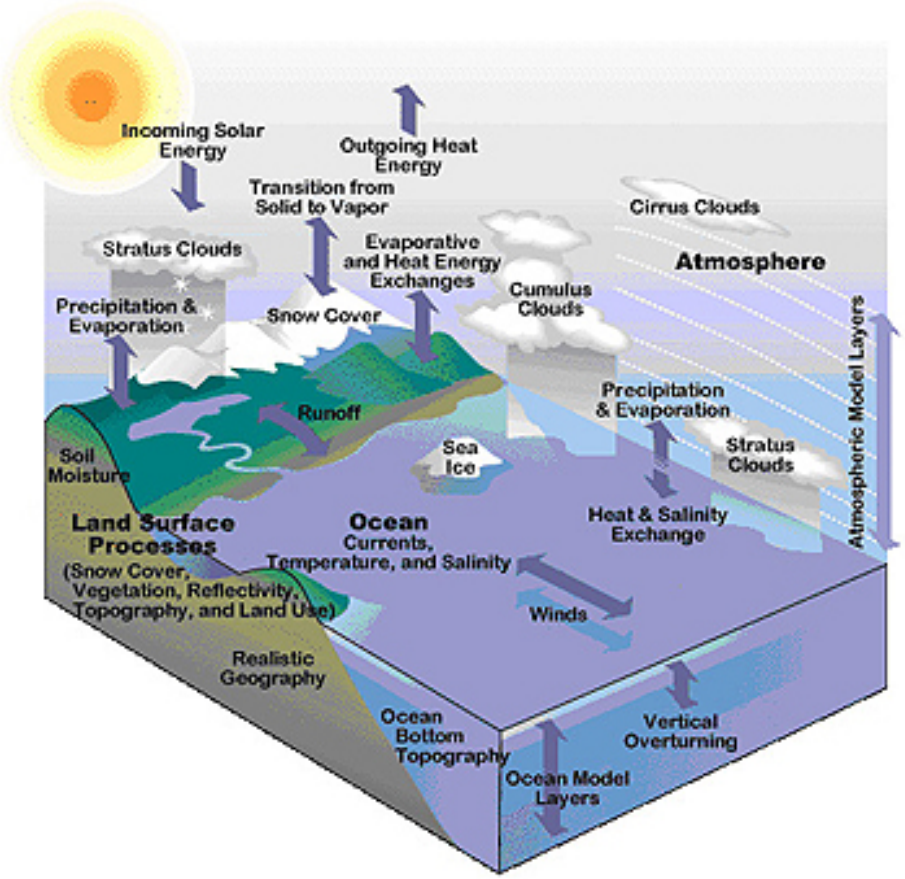
NCAR Correctnesss Workshop, Nov 10, 2023

The National Center for Atmospheric Research

- Boulder, Colorado
- funded by the National Science Foundation (NSF)
- *“to understand the behavior of the atmosphere and the related Earth system”*

- Cheyenne, WY
- ~20 petaflop HPE Cray EX cluster
- 323,712 processor cores
- 2,488 compute nodes with 128 AMD Milan cores per node
- 82 nodes w/4 NVIDIA A100 GPUs each
- HPE Slingshot high-speed interconnect

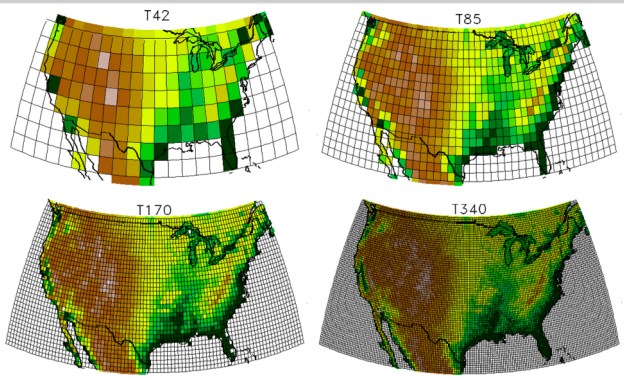
Community Earth System Model™ (CESM)

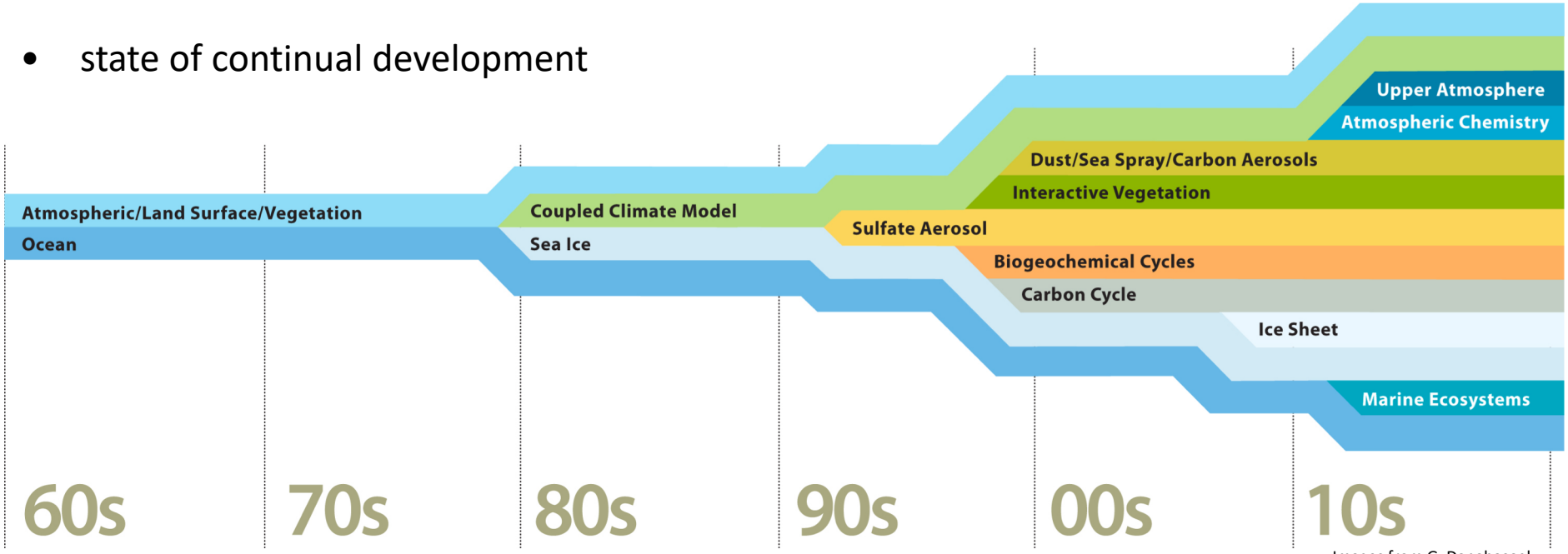


- models complex processes in the atmosphere, ocean, land, sea ice, glaciers, and rivers
- past, present and future climate states
- interdisciplinary collaborative effort (led by NCAR)
- widely-used:
  - ~5200 users just in the forums!
  - 492 downloads (clones) in the last 14 days
  - >5000 closed PR requests (across components)

CESM code base

- large Fortran code (~2.5M lines)
- 30+ years of code (modern and not-so-much)
- > 13,000 subroutines and >3,000 functions
- laptops, HPC systems, the cloud
- state of continual development





Images from G. Danabasoglu

Need for Software Quality Assurance

Insure that changes during the CESM development life cycle do not adversely affect the results!

- port to new environment (e.g., different institution)
- compiler changes
- code modifications (e.g., optimizations)
- heterogeneous computing
- …

Bit-for-Bit (BFB) Reproducibility?

CESM is deterministic and results are BFB reproducible ***if:***

same software version,
same compiler and flags,
same MPI,
same parameters settings,
same initial conditions,
same random number generator,
same hardware,…

not typically the case!

Bit-for-Bit (BFB) Reproducibility?

CESM is deterministic and results are BFB reproducible ***if:***

same software version,
same compiler and flags,
same MPI,
same parameters settings,
same initial conditions,
same random number generator,
same hardware,…

not typically the case!

Too restrictive!

- optimizing code
- new hardware technologies
- compiler flags (-O3)

Motivation

original “accepted” CESM data

Changes to hardware
or software environment
or CESM code

“new” CESM data

Key question: If $X \neq \tilde{X}$ is the output still correct?

Motivation

X

original "accepted" CESM data

Changes to hardware
or software environment
or CESM code

$\tilde{X}$

"new" CESM data

Key question: If $X \neq \tilde{X}$ is the output still correct?

Does the new data still represent the same climate?
Or is it "climate-changing"?

Evaluating the difference

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Challenge: there is no clear definition of "climate-changing"

Past approach: compare long simulations (~400 years)

- climate expertise required
- subjective
- computationally expensive
- time consuming (hundreds of output variables!)

Evaluating the difference

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Challenge: there is no clear definition of "climate-changing"

Past approach: compare long simulations (~400 years)

- climate expertise required
- subjective
- computationally expensive
- time consuming

Need an automated tool!

- easy-to-use
- objective

Towards an easy-to-use objective automated tool …

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Towards an easy-to-use objective automated tool …

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Let's reframe the problem!

Our new approach: Ensemble Consistency Test

New question: Is the new data *statistically distinguishable* from "accepted" data?

Our new approach: Ensemble Consistency Test

New question: Is the new data *statistically distinguishable* from "accepted" data?

Approach: evaluate in the context of the climate model's ***variability***

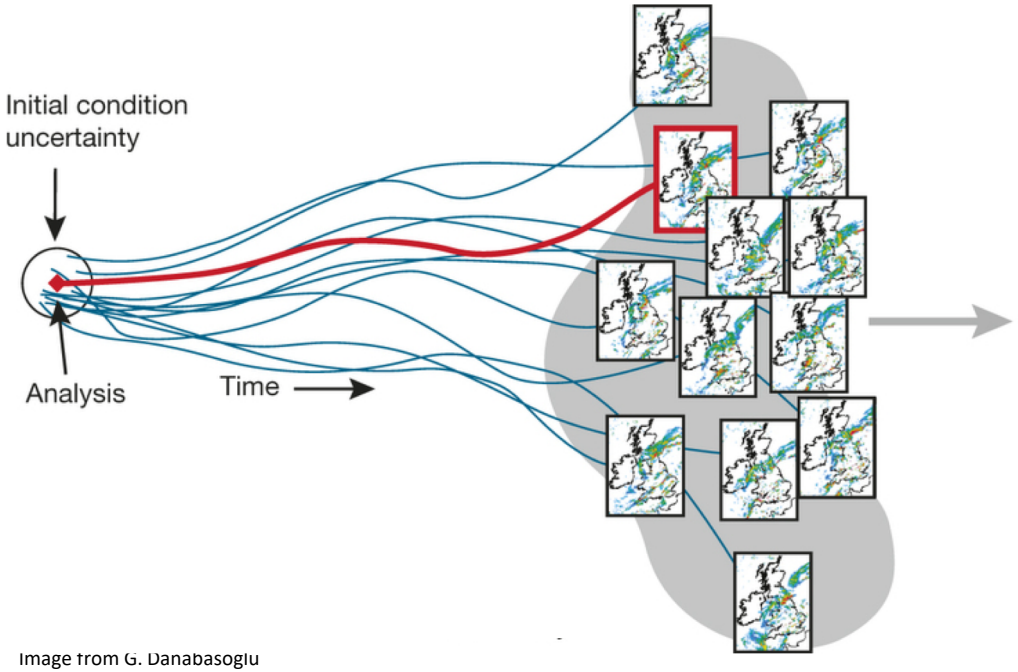


Image from G. Danabasoglu

i.e., evaluate new data in the context of an ensemble of "accepted" CESM runs

Our new approach: Ensemble Consistency Test

New question: Is the new data *statistically distinguishable* from "accepted" data?

Approach: evaluate in the context of the climate model's ***variability***

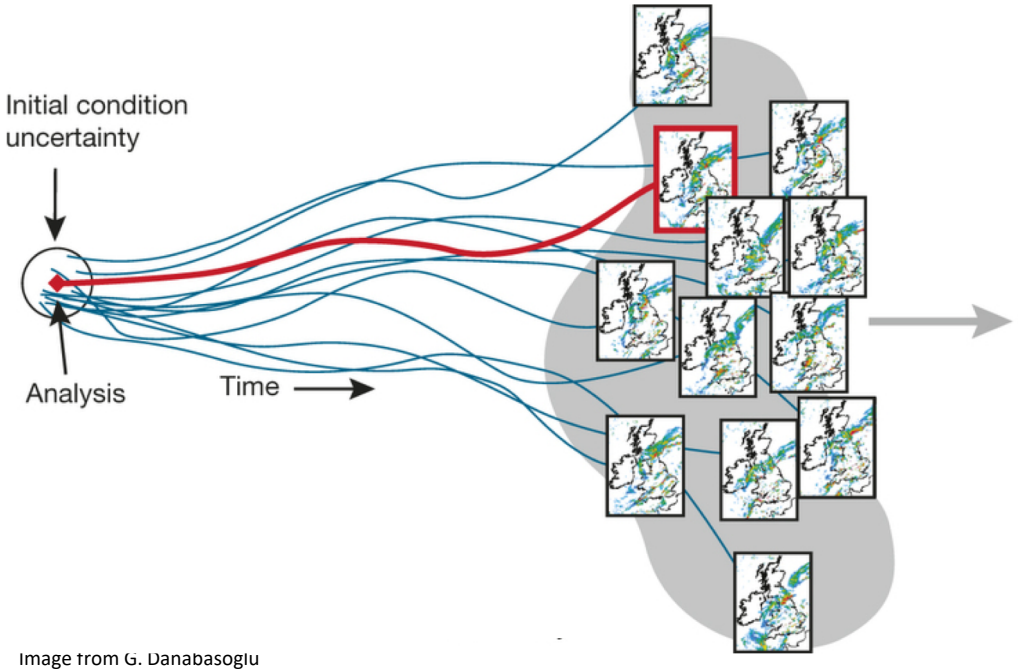


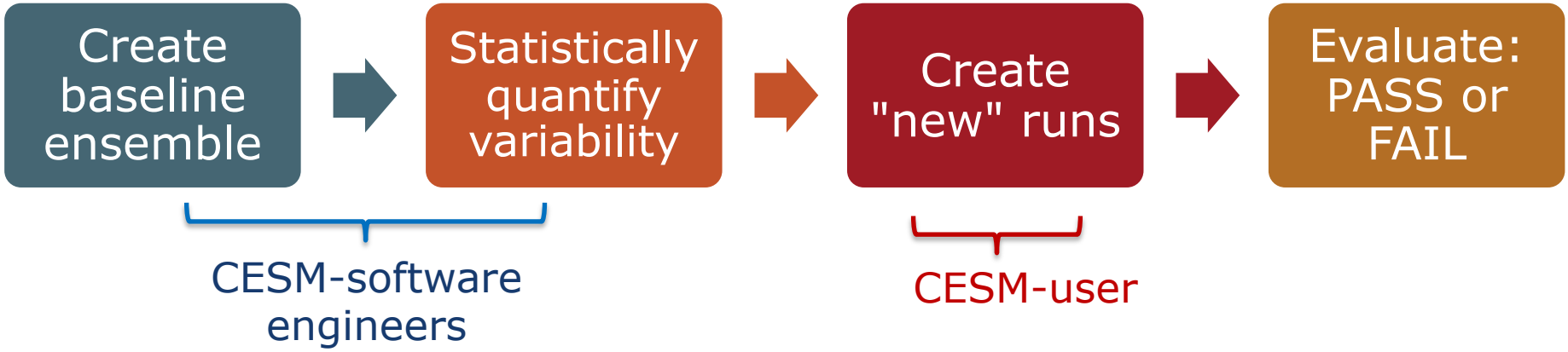
Image from G. Danabasoglu

i.e., evaluate new data in the context of an ensemble of "accepted" CESM runs

High dimensionality is a key issue: 200+ variables ….

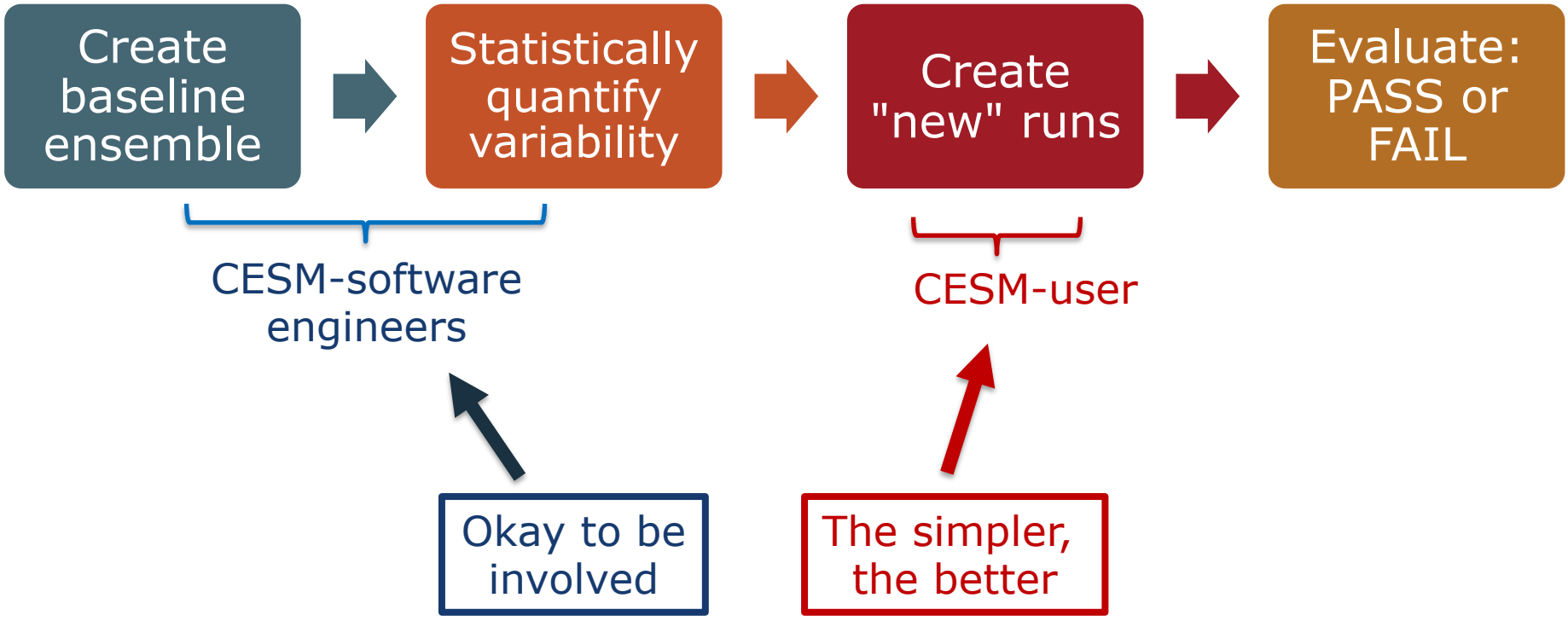
Ensemble Consistency Test (ECT)

Overview:



Ensemble Consistency Test (ECT)

Overview:



Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature

Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

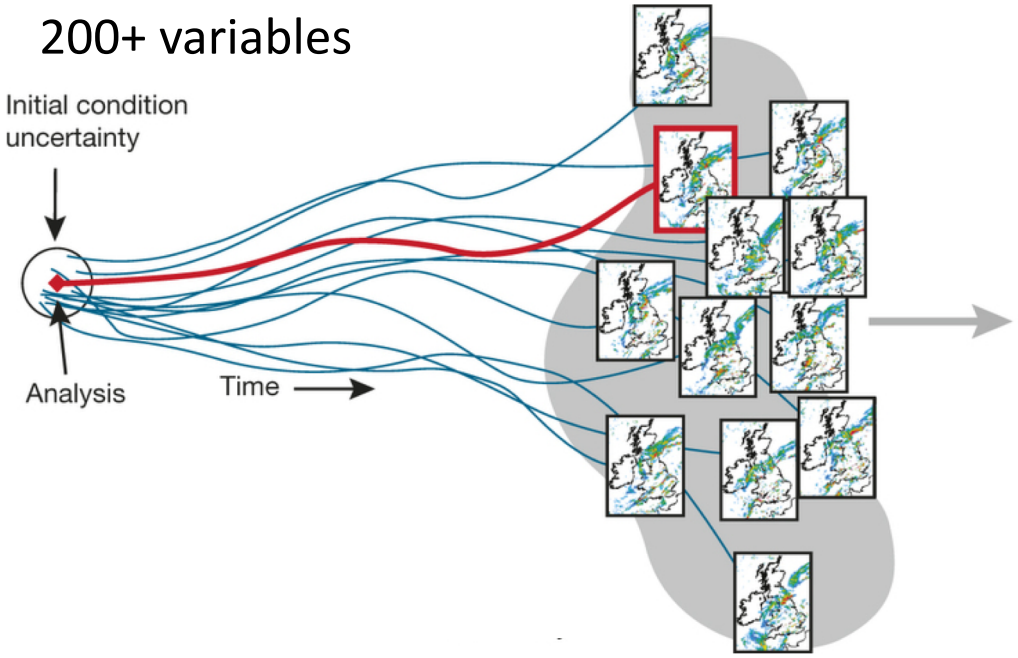
- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature

Many other options to create ensemble; and they can matter!

Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

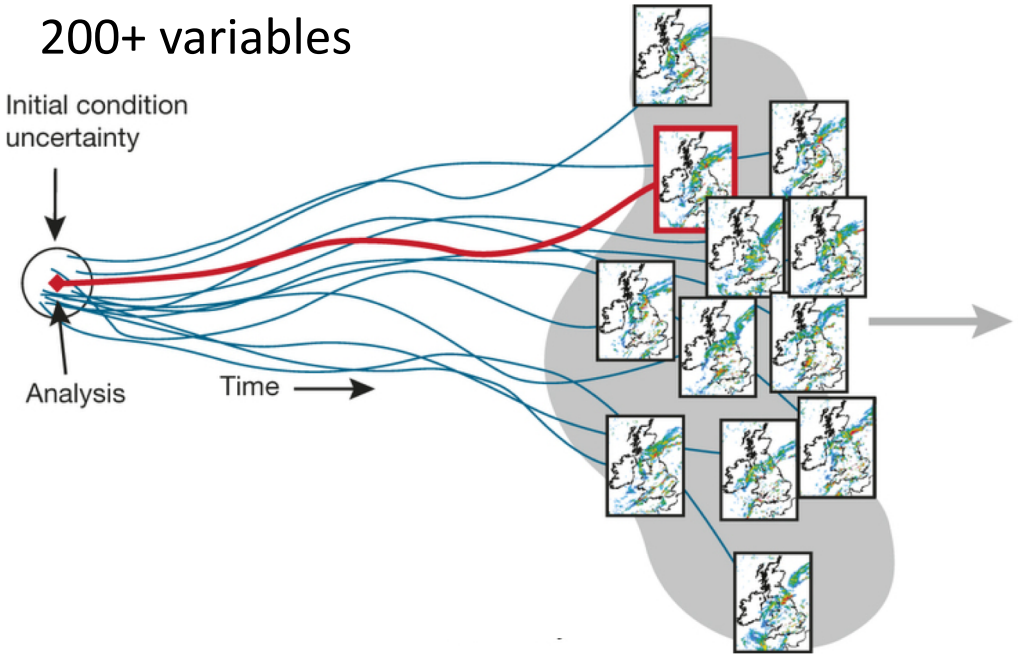
- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- one-year simulations
- 200+ variables



Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- one-year simulations
- 200+ variables



The mean of each field is all that is needed…

Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- one-year simulations
- 200+ **globally-averaged** variables

Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- one-year simulations
- 200+ globally-averaged variables

Compare variable value in "new" run to its ensemble distribution:

- many variables are highly correlated!
- difficult to make pass/fail choices based on variables one-by-one

Creation of and comparison with ensemble

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- one-year simulations
- 200+ globally-averaged variables

Compare variable value in "new" run to its ensemble distribution:

- many variables are highly correlated!
- difficult to make pass/fail choices based on variables one-by-one

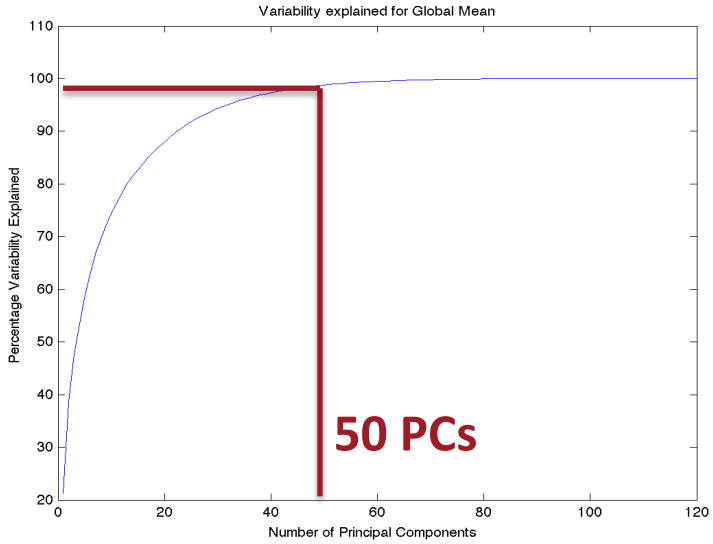
→ **use Principal Component Analysis (PCA)**

Quantify ensemble variability

New testing tool based on Principal Component Analysis (PCA):

- standardize variables (different scales)
- project data into orthogonal space
  (orthogonalize data in the direction of maximized variability…)
- resulting linear combinations of variables (scores) are used for the ensemble distribution
- use enough scores to represent
  most of the variance

compare scores from new runs to distribution of scores from ensemble



Hypothesis Testing based on Principal Components

Key: picks up "correctness" of relationships between variables

null hypothesis (H_0): the new climate simulations come from the same distribution as the ensemble simulations.

ECT issues a pass or fail, and must balance:

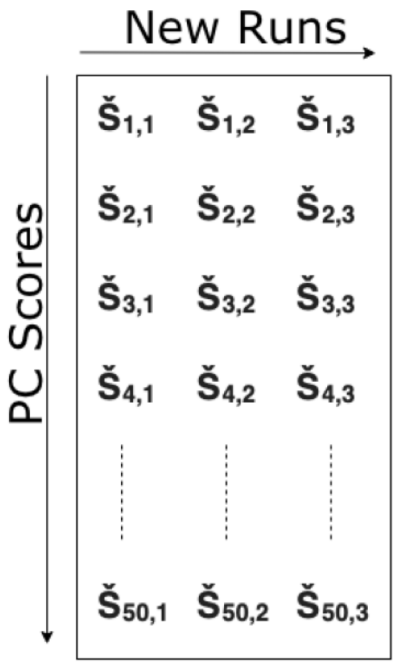
- false positive rate: probability of falsely rejecting H_0 when it is true
- power: probability of correctly rejecting H_0 when it is false

Ideally:

- false positive rate is as low as possible
- power is as high as possible

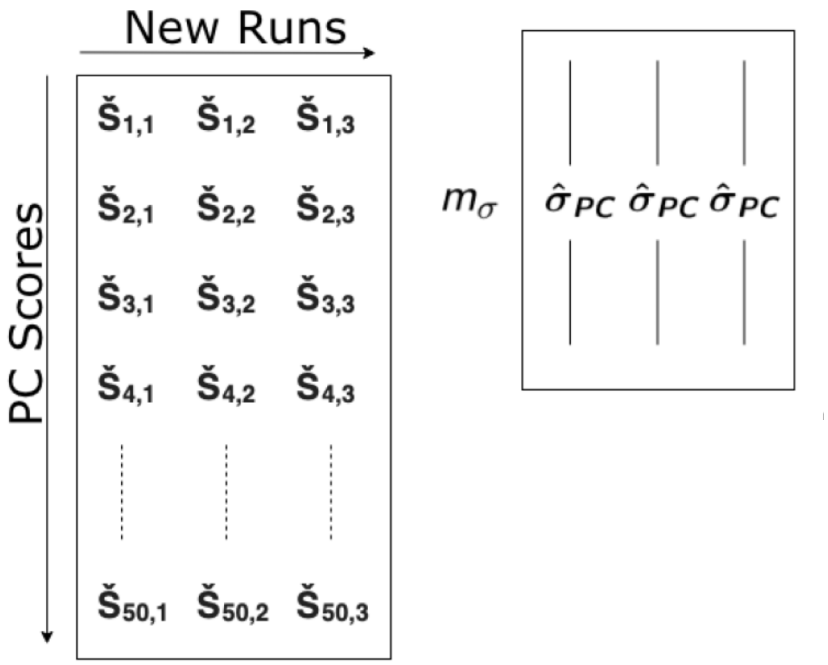
ECT Procedure

$$N_{new} = 3 \text{ and } N_{PC} = 50$$



ECT Procedure

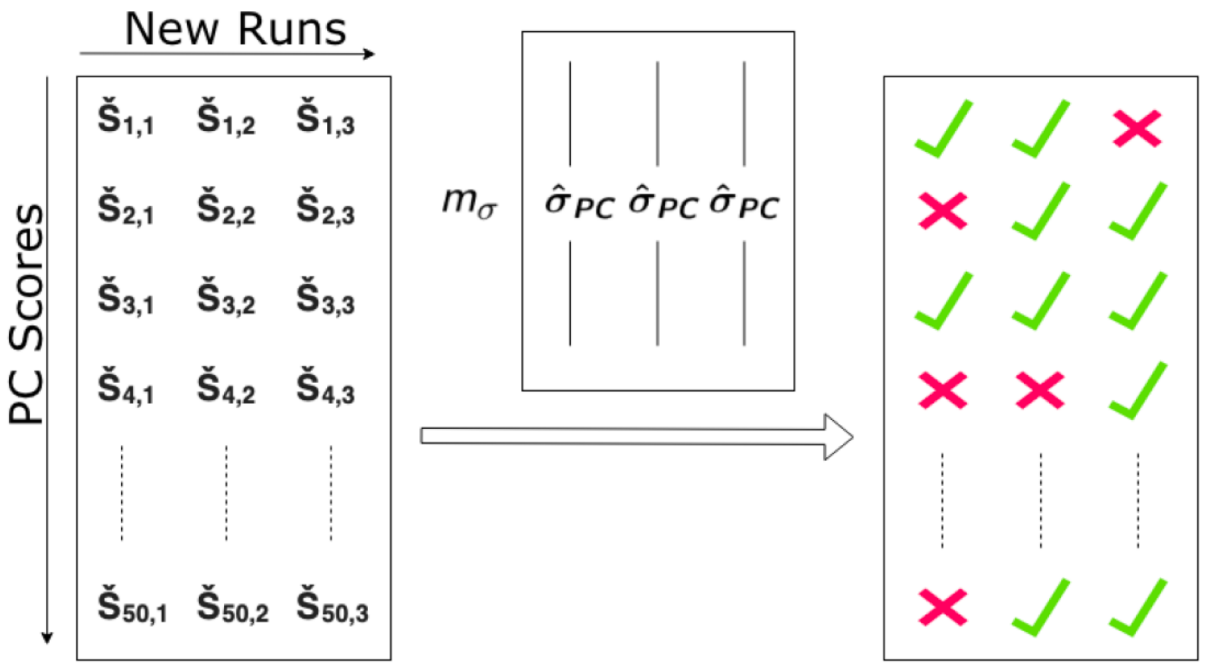
$$N_{new} = 3 \text{ and } N_{PC} = 50$$



$$m_\sigma = 2$$

ECT Procedure

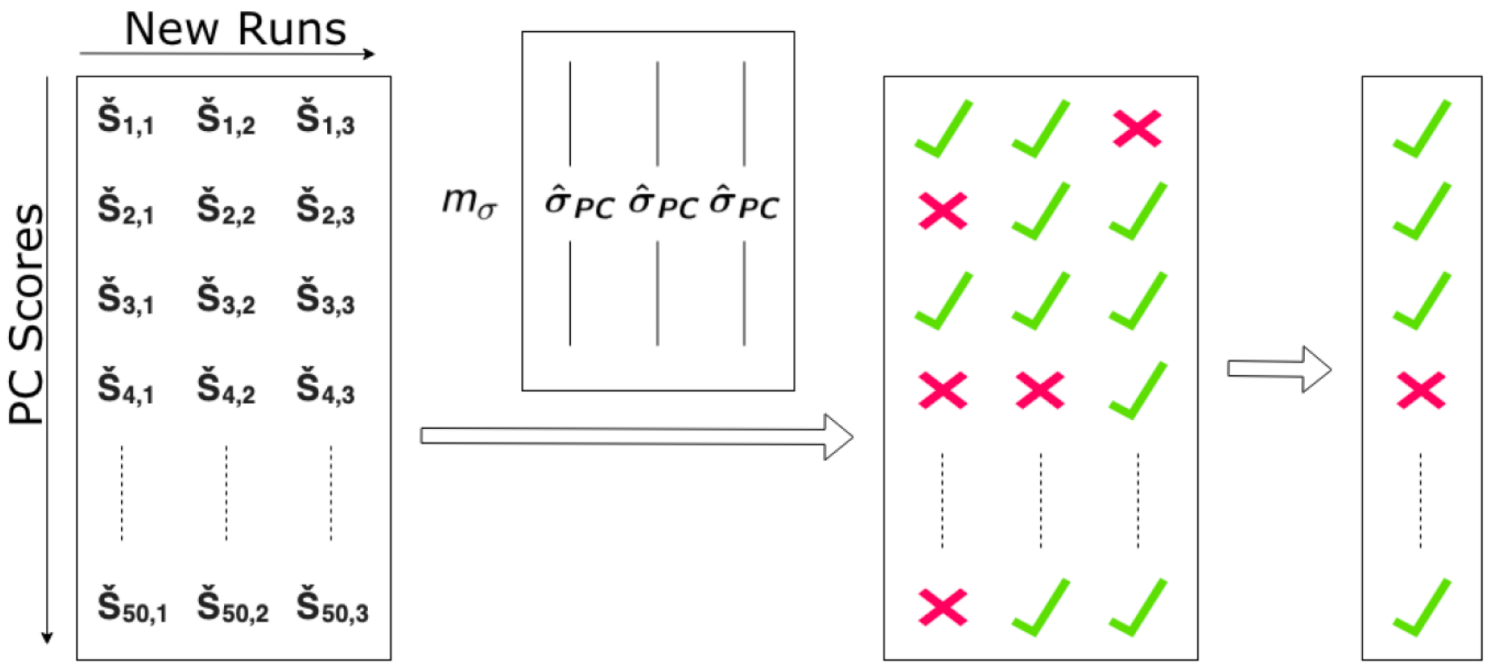
$N_{new} = 3$ and $N_{PC} = 50$



$m_\sigma = 2$

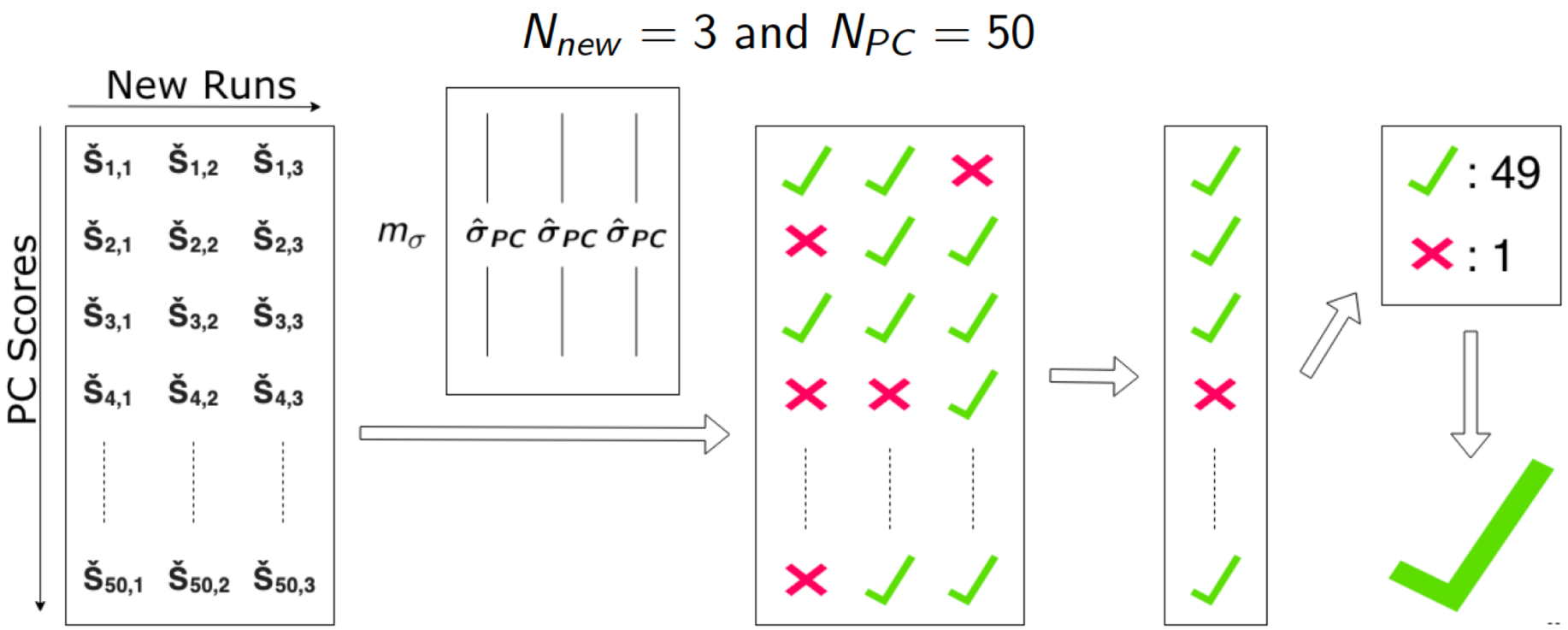
ECT Procedure

$N_{new} = 3$ and $N_{PC} = 50$



$m_\sigma = 2,\ N_{runFails} = 2$

ECT Procedure



How well does CAM-ECT work?

Lots of testing:

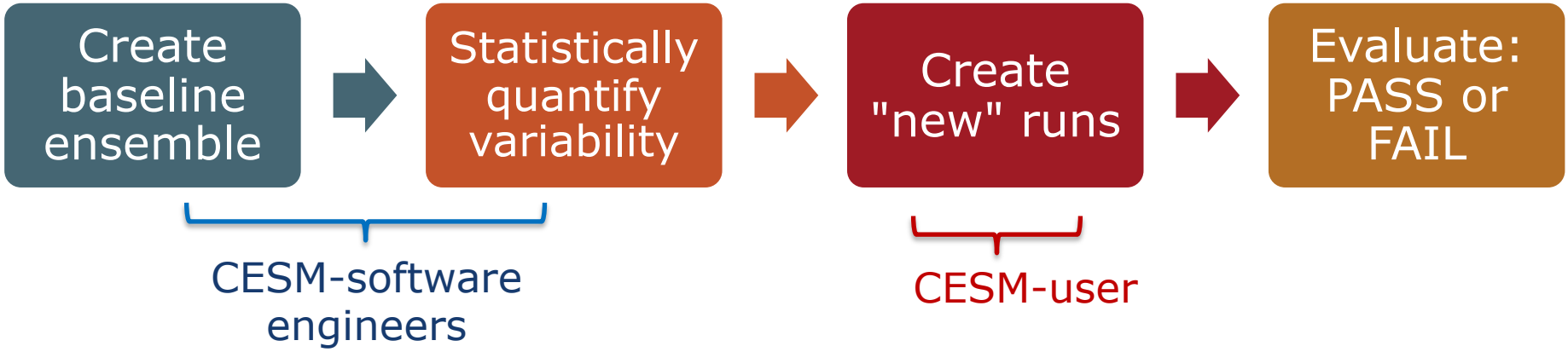
- modifications *expected* to be climate-changing *fail*
  - e.g. relative humidity, dust emissions, CO_2 levels
- modifications *not expected* to be climate changing *pass*
  - e.g., threads, -O0, compiler version, code rearrangement

In-use:

- CESM port-verification and code optimization
- uncovered errors in code and hardware
- works extremely well in practice - *hard to find any "real error" it doesn't catch!*

Ensemble Consistency Test (ECT)

Overview:

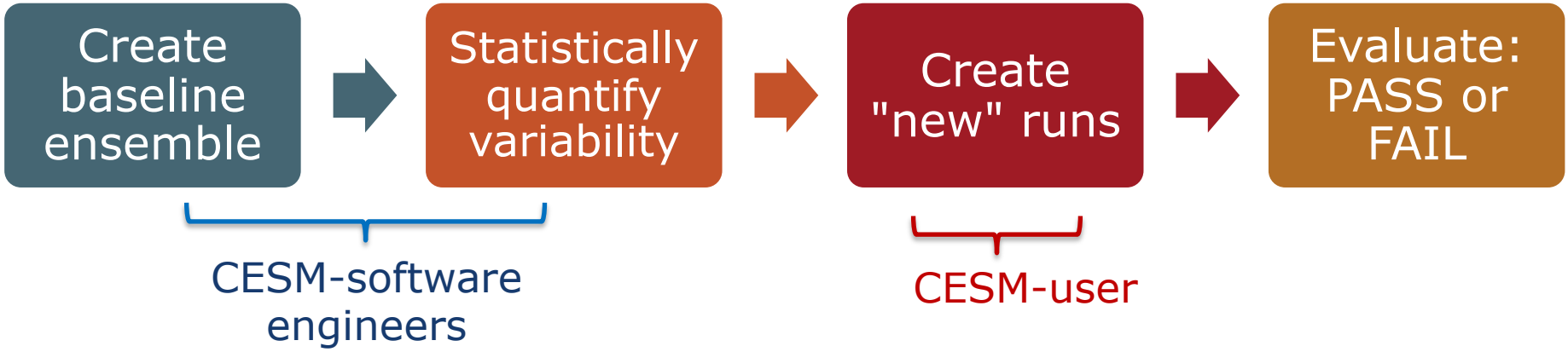


Highlights:

- automated Python tool
- objective, user-friendly
- rapid feedback for model developers
- suite of tools (atmosphere, land, ocean)

Ensemble Consistency Test (ECT)

Overview:



Highlights:

- automated Python tool
- objective, user-friendly
- rapid feedback for model developers
- suite of tools (atmosphere, land, ocean)

climate-modeling expertise is not required!

Do we really need year-long runs?

ECT works well… (using annual averages)

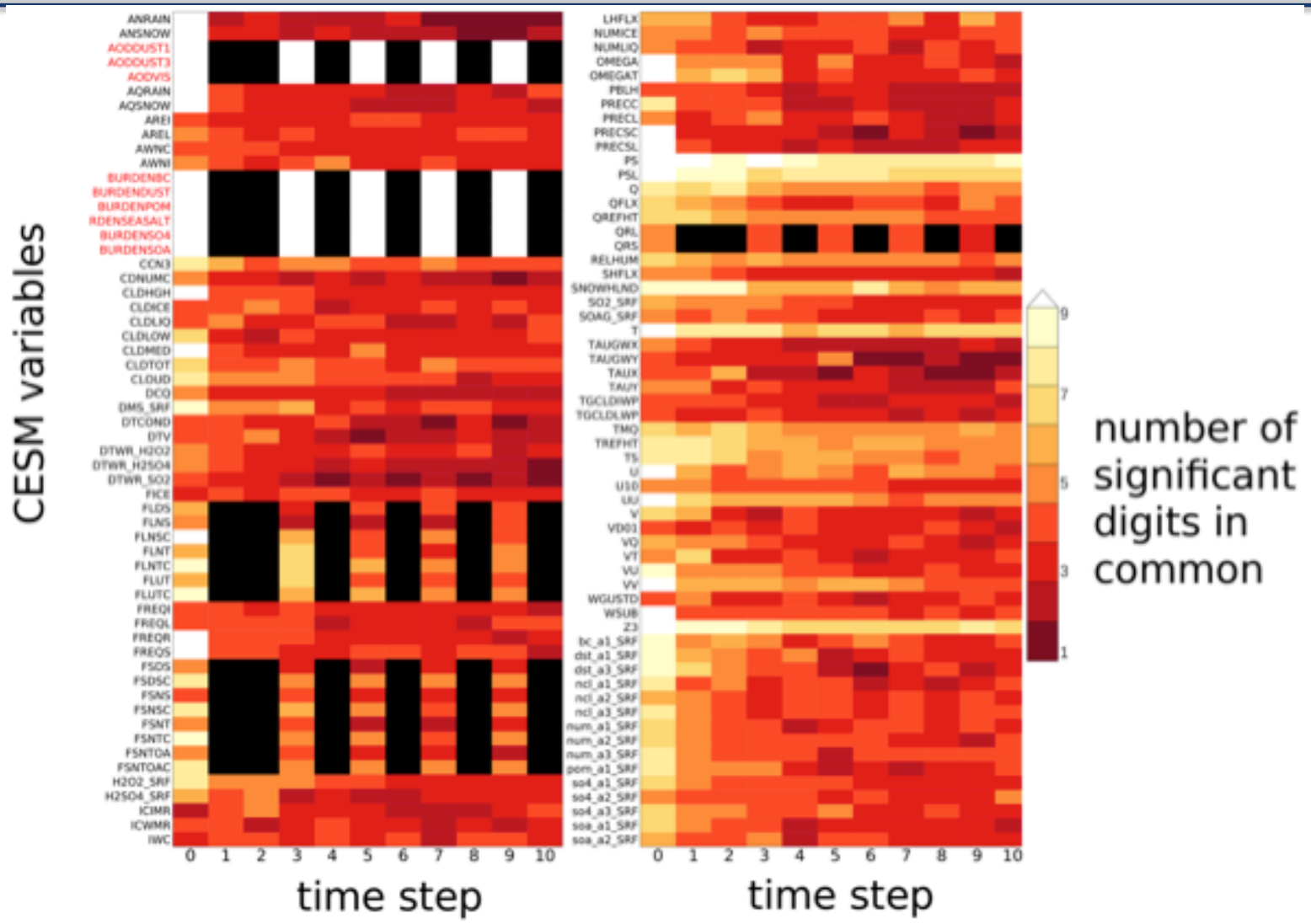
But could the ensemble simulations be shorter?

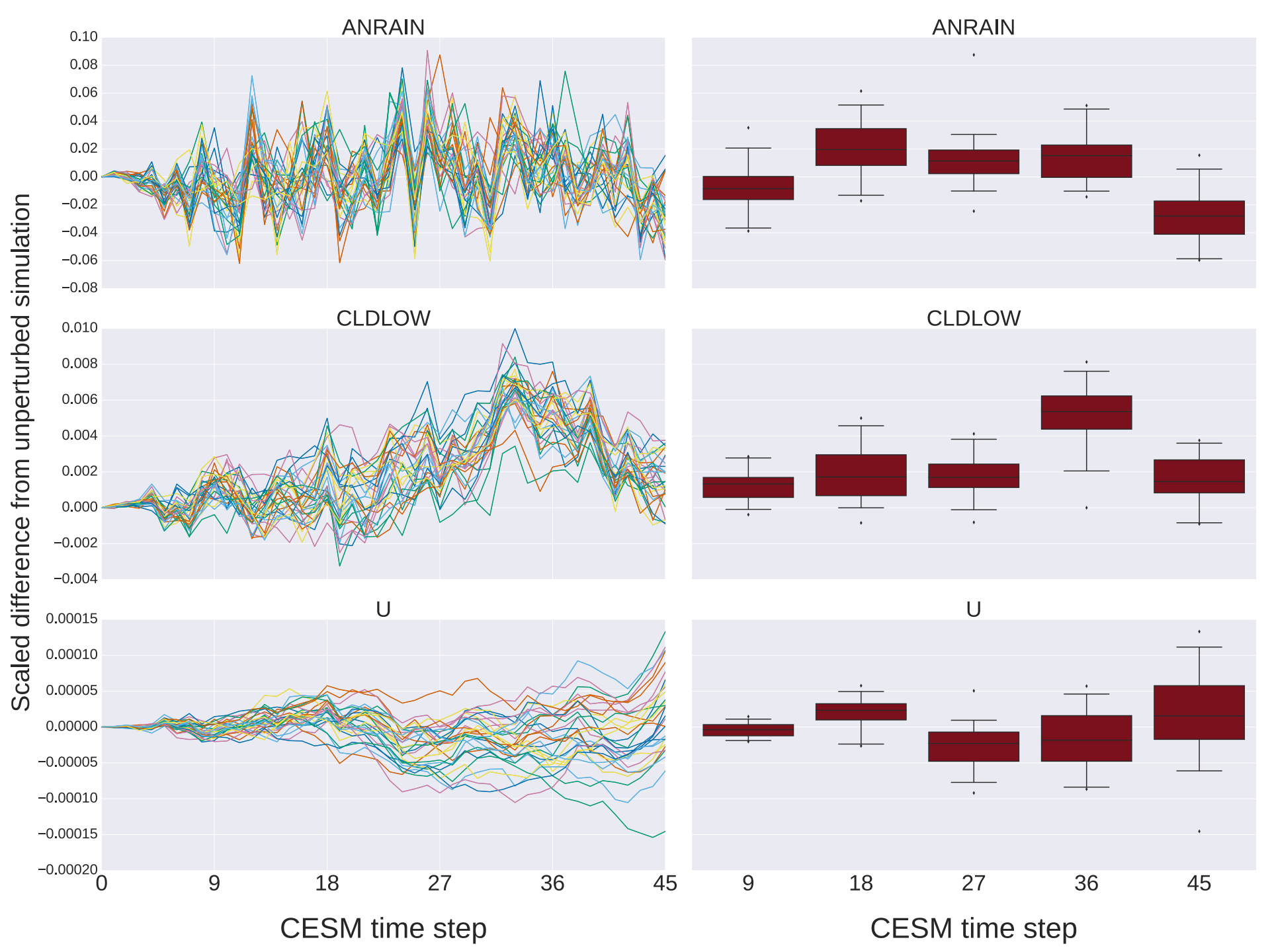
- shorter = cheaper
- larger ensemble sizes possible

How about just a small number of time steps?

- initial perturbations grow fast…
- too much spread?
- too little spread?

First 10 time steps: original vs. perturbed




ANRAIN
ANRAIN
CLDLOW
CLDLOW
U
U
Scaled difference from unperturbed simulation
CESM time step
CESM time step

UF-CAM-ECT ("ultra-fast")

- 9 time steps ~ 70x cheaper (NCAR machine ~ 1.5min)
- instantaneous values (sensitive to localized phenomenon)
- works surprisingly well!!

UF-CAM-ECT and CAM-ECT are *almost always* in agreement:

- compiler changes
- new machines
- minor code modifications
- CAM namelist alterations
- CLM modifications

UF-CAM-ECT ("ultra-fast")

- 9 time steps ~ 70x cheaper (NCAR machine ~ 1.5min)
- instantaneous values (sensitive to localized phenomenon)
- works surprisingly well!!

UF-CAM-ECT and CAM-ECT are *almost always* in agreement:

- compiler changes
- new machines
- minor code modifications
- CAM namelist alterations
- CESM modifications

Lots of tests!

Very difficult to find counter-examples!

Counterexample 1: HYDRO-BASEFLOW

HYDRO-BASEFLOW: increase the soil hydrology baseflow rate coefficient (10,000x)

controls the amount of water drained from the soil

UF-CAM-ECT passes: change undetectable in CAM data at 9th time step (deep soil)

CAM-ECT fails: through the year the change propagates to the surface fluxes

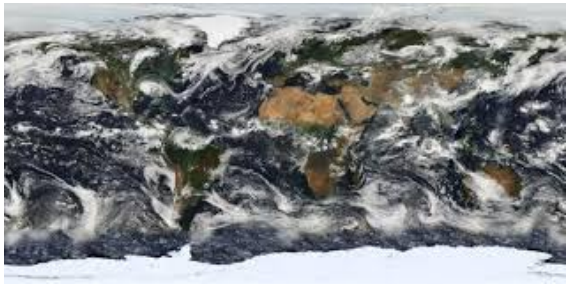
***Only very contrived/manufactured examples with this behavior

Counterexample 2: RAND

RAND: change the *pseudo random number generator* (PRNG) in the radiation module

UF-CAM-ECT fails: identifies a change in characteristics of cloud-related variables at 9th time step

CAM-ECT passes: we don't expect using a different PRNG to make a big difference in annual average …. *and they don't!*



UF-CAM-ECT and CAM-ECT

In practice:

- Use UF-CAM-ECT by default (cheaper & disagreements are rare)
- Use CAM-ECT when UF-CAM-ECT returns an *unexpected* fail

Remarks:

- nice option when bit-for-bit reproducibility is not possible…
- objective, user-friendly
- uncovered multiple errors in code and hardware
- can detect changes in variable relationships (PCA)

Variants for other components….

A highly accurate test leads to new challenges …

Specific motivating instance: porting to a new HPC system: fail!

- several months, 10+ people, major headache!

A highly accurate test leads to new challenges …

Specific motivating instance: porting to a new HPC system: fail!

- several months, 10+ people, major headache!

⟶ eventually identified cause: inconsistency with FMA (Fused Multiply-Add)

CESM Challenge: size and complexity of (Fortran) code

A highly accurate test leads to new challenges …

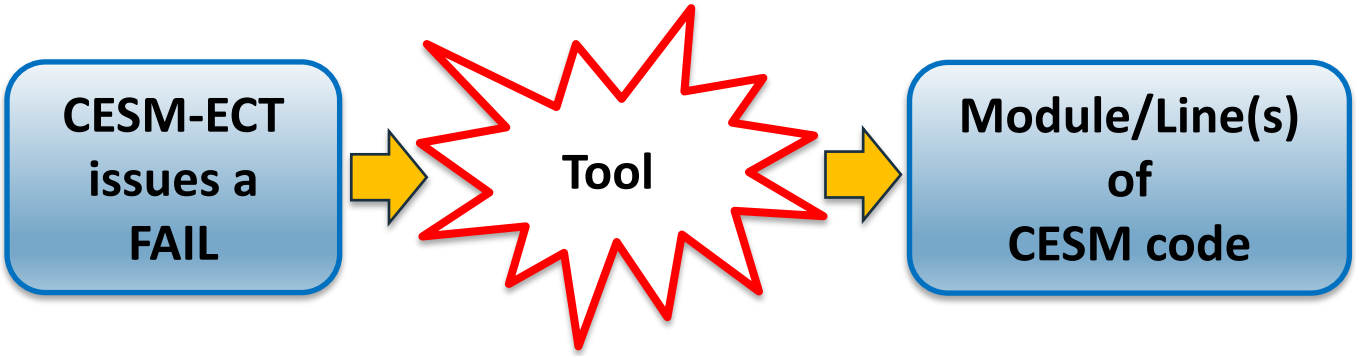
Specific motivating instance: porting to a new HPC system: fail!

- several months, 10+ people, major headache!

⟶ eventually identified cause: inconsistency with FMA (Fused Multiply-Add)

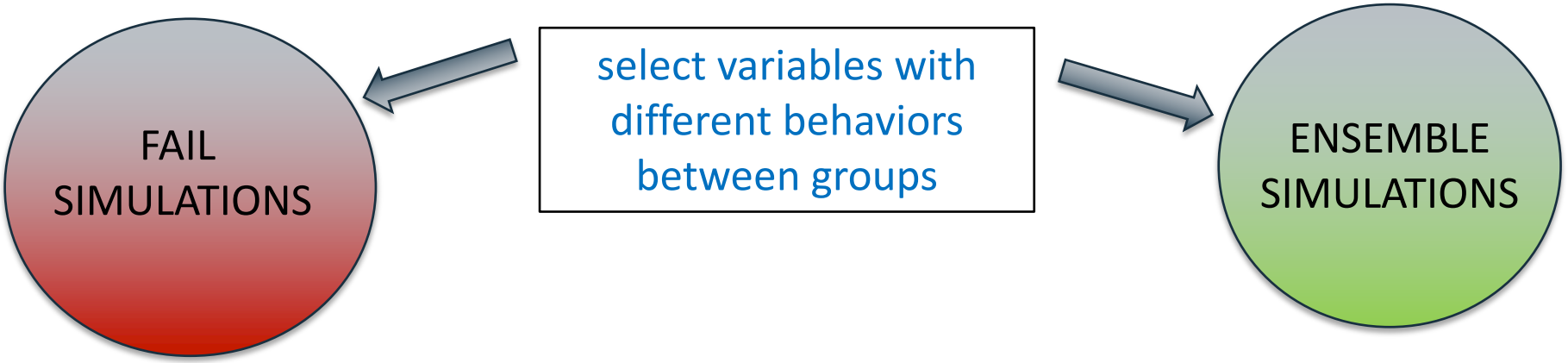
CESM Challenge: size and complexity of (Fortran) code

Goal: Give helpful information when CESM-ECT issues a "fail" !



First step: identify affected variables

Identify affected output variables (at < 9 timesteps):



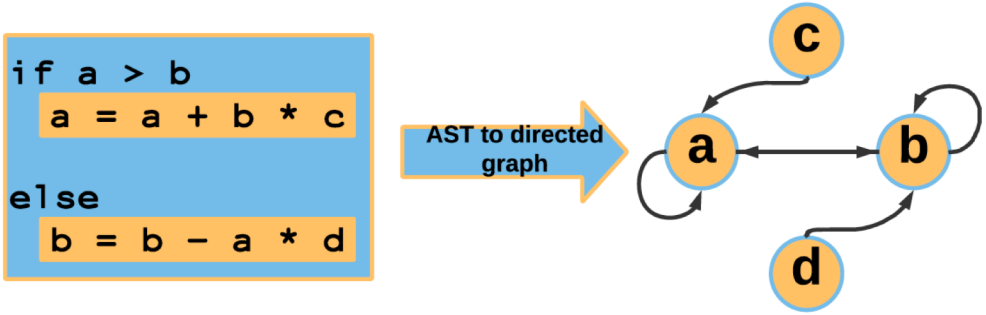
A few options:

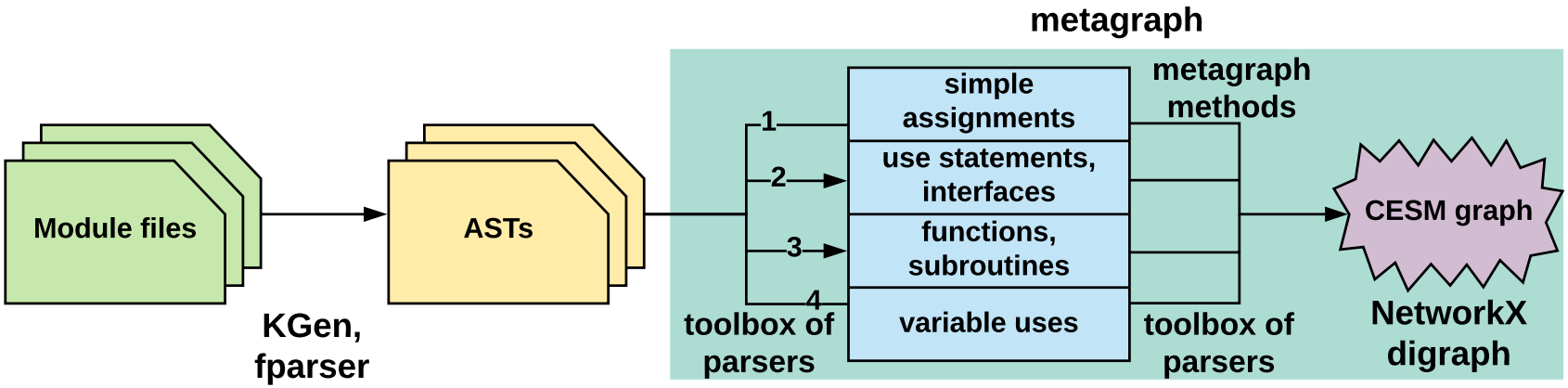
- only a few variables with differences at t = 1 ?
- compare variable distributions
- logistic regression w/regularization (lasso)

Next: convert source code to directed graph

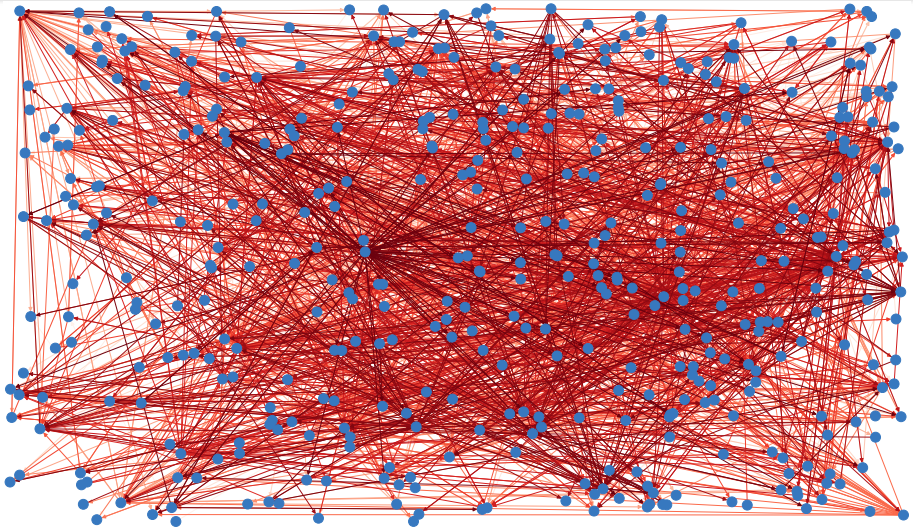
Convert source code to graph:

- prune source code (code coverage tool)
- build Abstract Syntax Tree (AST) => *parse the CESM source!*
- convert ASTs into a directed graph

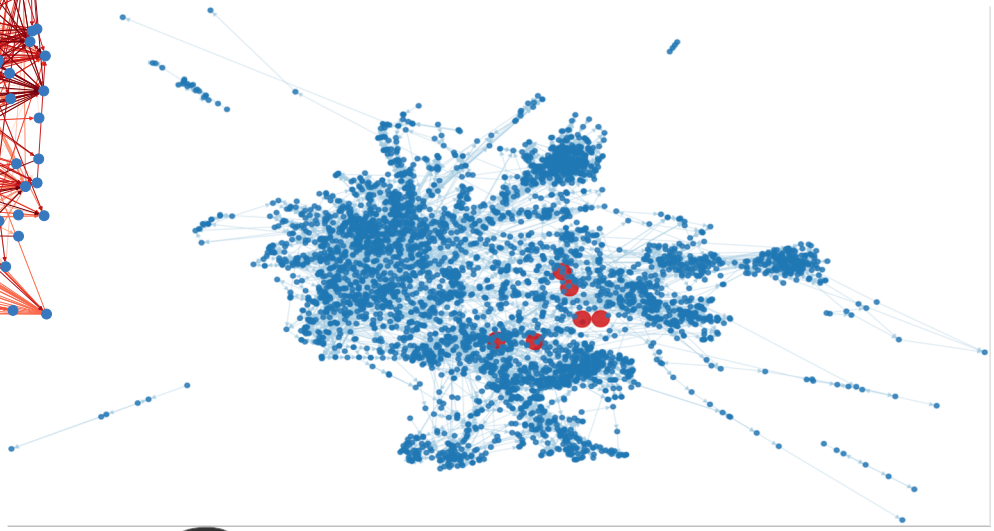




Next next: reduce to relevant subgraph

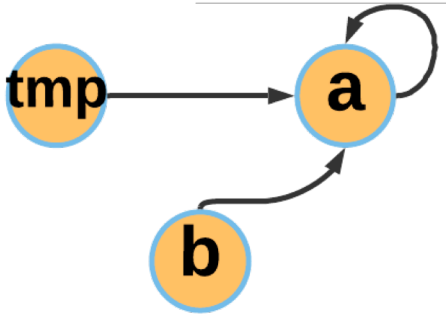


CESM modules



Use affected variables to get induced subgraph of paths (i.e., static slicing):

```
!tmp = 1.e-2r_8
tmp = 1.e2r_8
a = b + a*tmp
```

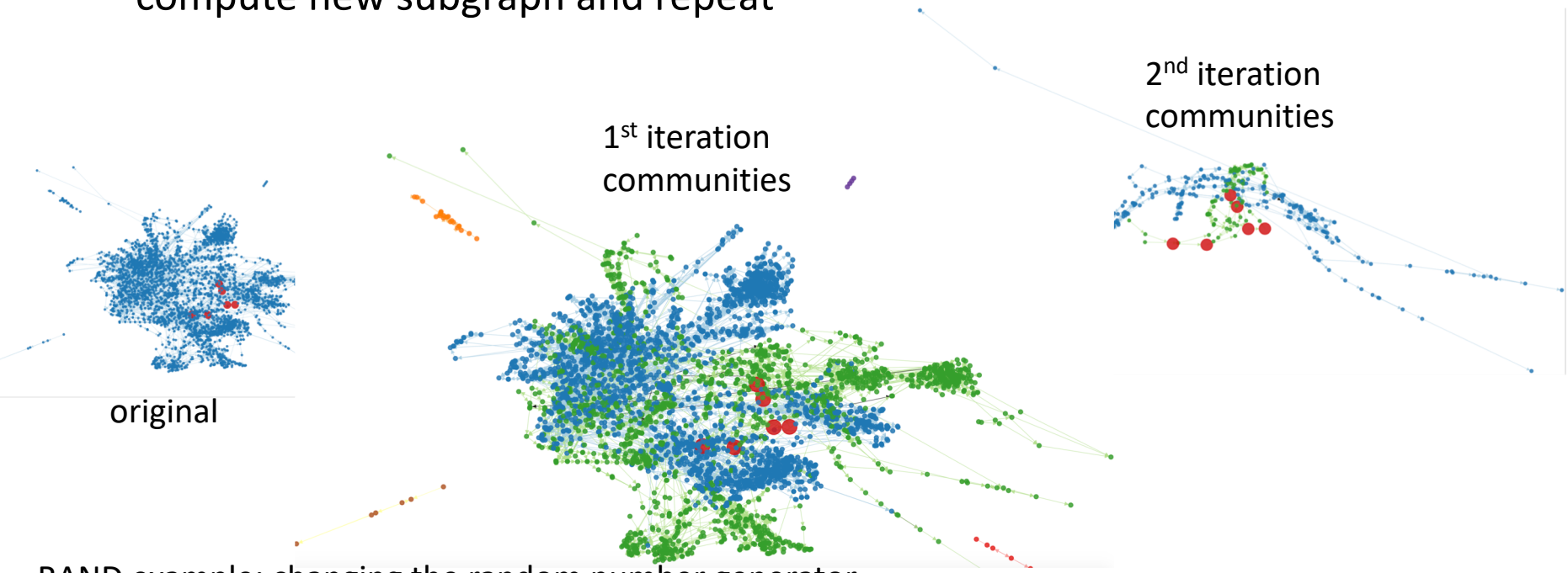


RAND example

Digging deep: iteratively narrowing the search

Value instrumentation is expensive – find a few "good" locations:

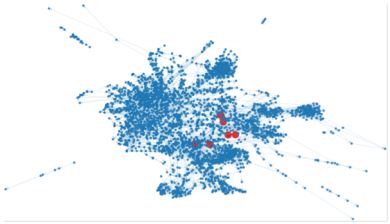
- divide subgraph into communities (via Girvan-Newman)
- use centrality within each community to rank nodes' importance
- instrument *most important* nodes in each community
- compute new subgraph and repeat



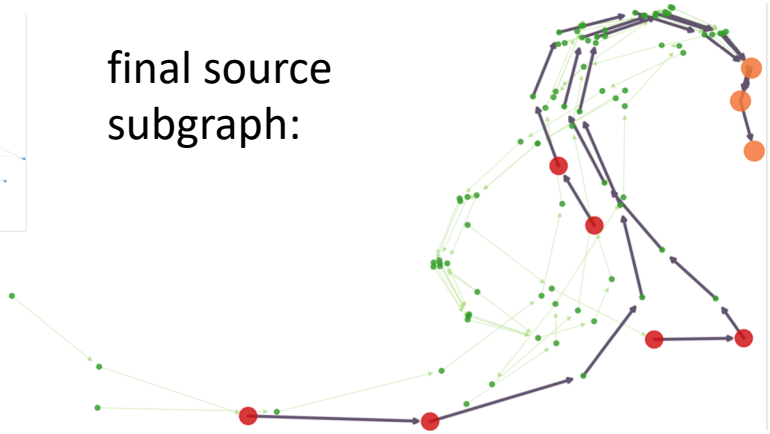
RAND example: changing the random number generator

…

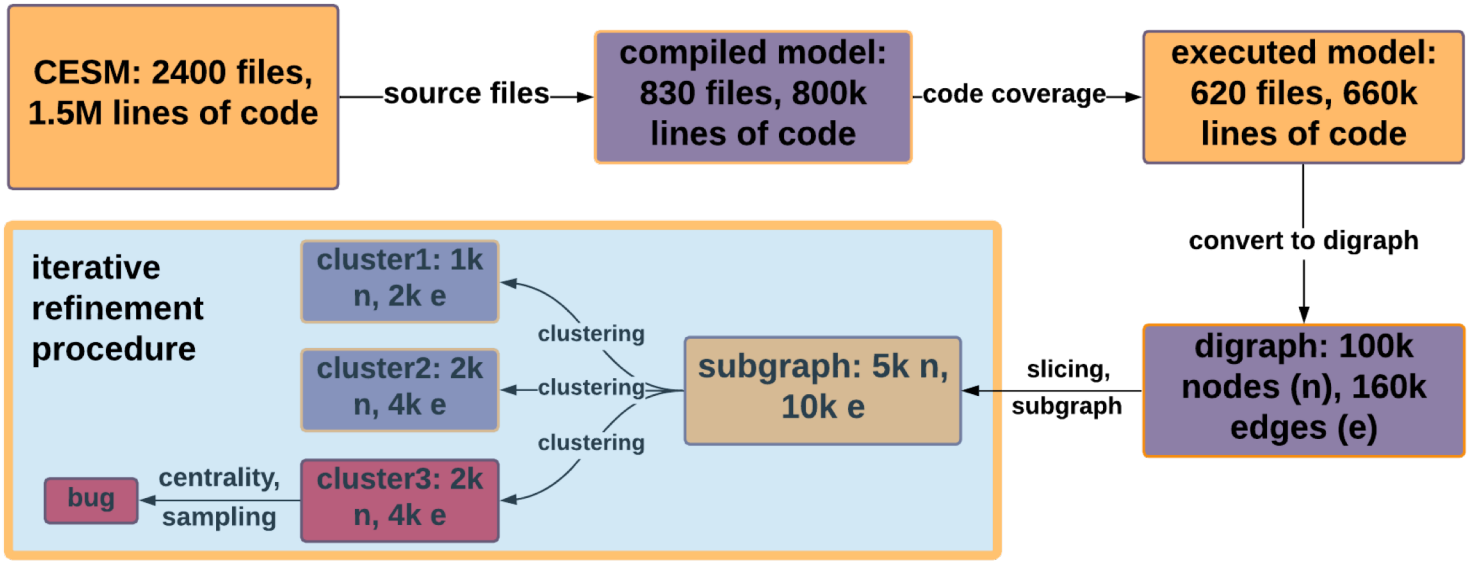
Promising, but still a long way to go!



final source subgraph:



Next step:
- implement iterative refinement algorithm



Shifting gears!

Looking at the statistical details of the ECT …

Derivation of theoretical false positive rate:

The structure of the CESM-ECT pass/fail scheme is such that we can analytically derive the theoretical false positive rate under certain distributional assumptions.

- We assume the test data come from the same distribution as the ensemble, and then calculate the probability of the CESM-ECT producing a failing result.

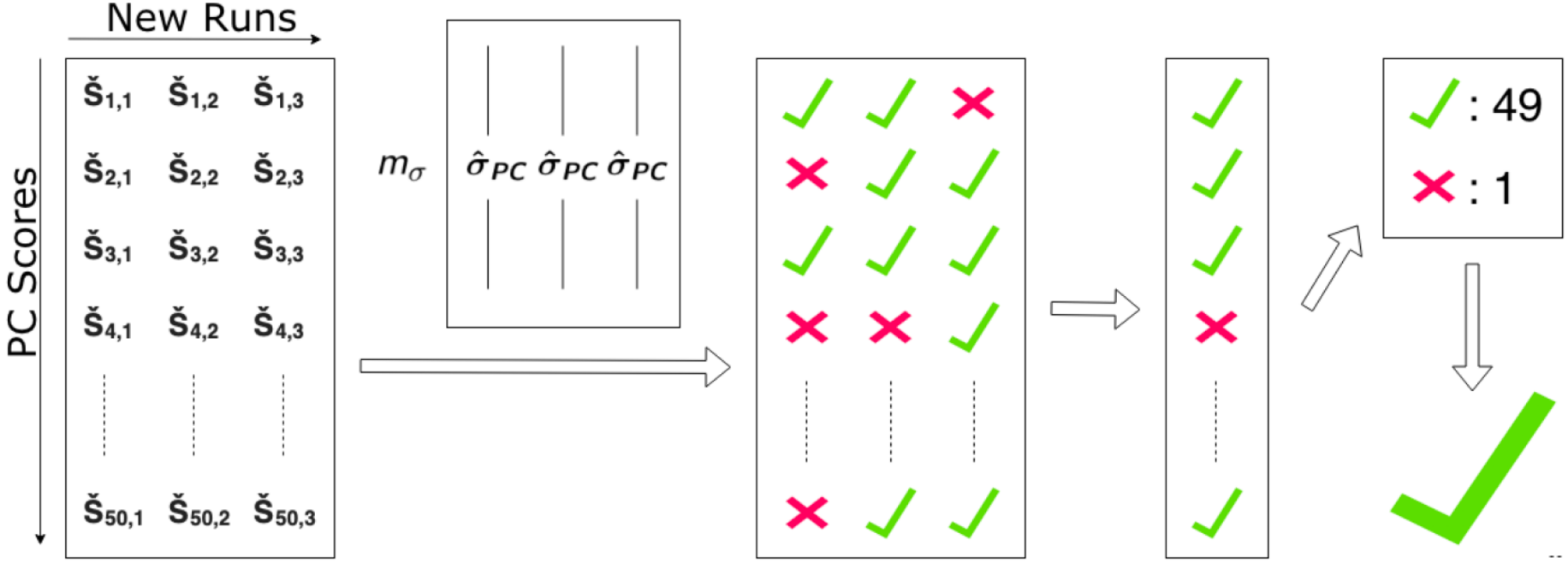
- We think of the entire CESM-ECT as a Bernoulli random variable that takes in test data and outputs either a 0 if the test data passes or 1 if the test data fails.

- We break this Bernoulli random variable down into the composition of several “subrandom variables" that correspond to the individual steps of the pass/fail scheme.

Investigation of Statistical Properties of ECT

Recall the setup:

$N_{new} = 3$ and $N_{PC} = 50$



$m_\sigma = 2$, $N_{runFails} = 2$, and $N_{pcFails} = 3$

Theoretical derivation of false positive rate: PC scores

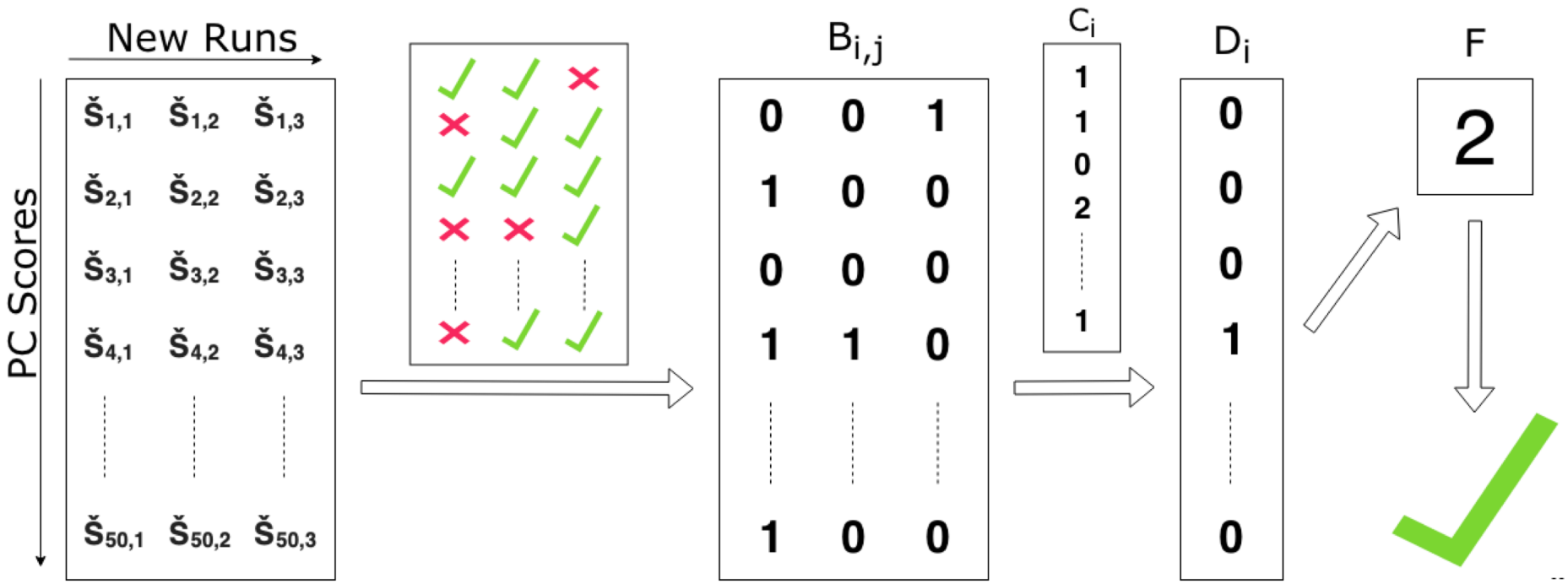
First, we consider the assessment of the individual PC scores. This step can be formulated as another Bernoulli random variable.

- Specifically, we define the random variable $B_{i,j} = 1$ if its respective PC score $\check{S}_{i,j} \notin (-2\,\hat{\sigma}_{PC,j}, 2\,\hat{\sigma}_{PC,j})$ and $B_{i,j} = 0$ if its respective PC score $\check{S}_{i,j} \in (-2\,\hat{\sigma}_{PC,j}, 2\,\hat{\sigma}_{PC,j})$.
- And calculate $p_B = P(B_{i,j} = 1)$:

$$\begin{aligned} p_B &= P(B_{i,j} = 1) \\ &= P(\check{S}_{i,j} \notin (-2\,\hat{\sigma}_{PC,i}, 2\,\hat{\sigma}_{PC,i})) \\ &= P(Z \notin (-2, 2)) \\ &= 2\ P(Z \leq -2) \\ &\approx 0.04550, \end{aligned}$$

where $Z \sim N(0, 1)$ is the standard normal random variable.

The ECT scheme viewed as a series of RVs

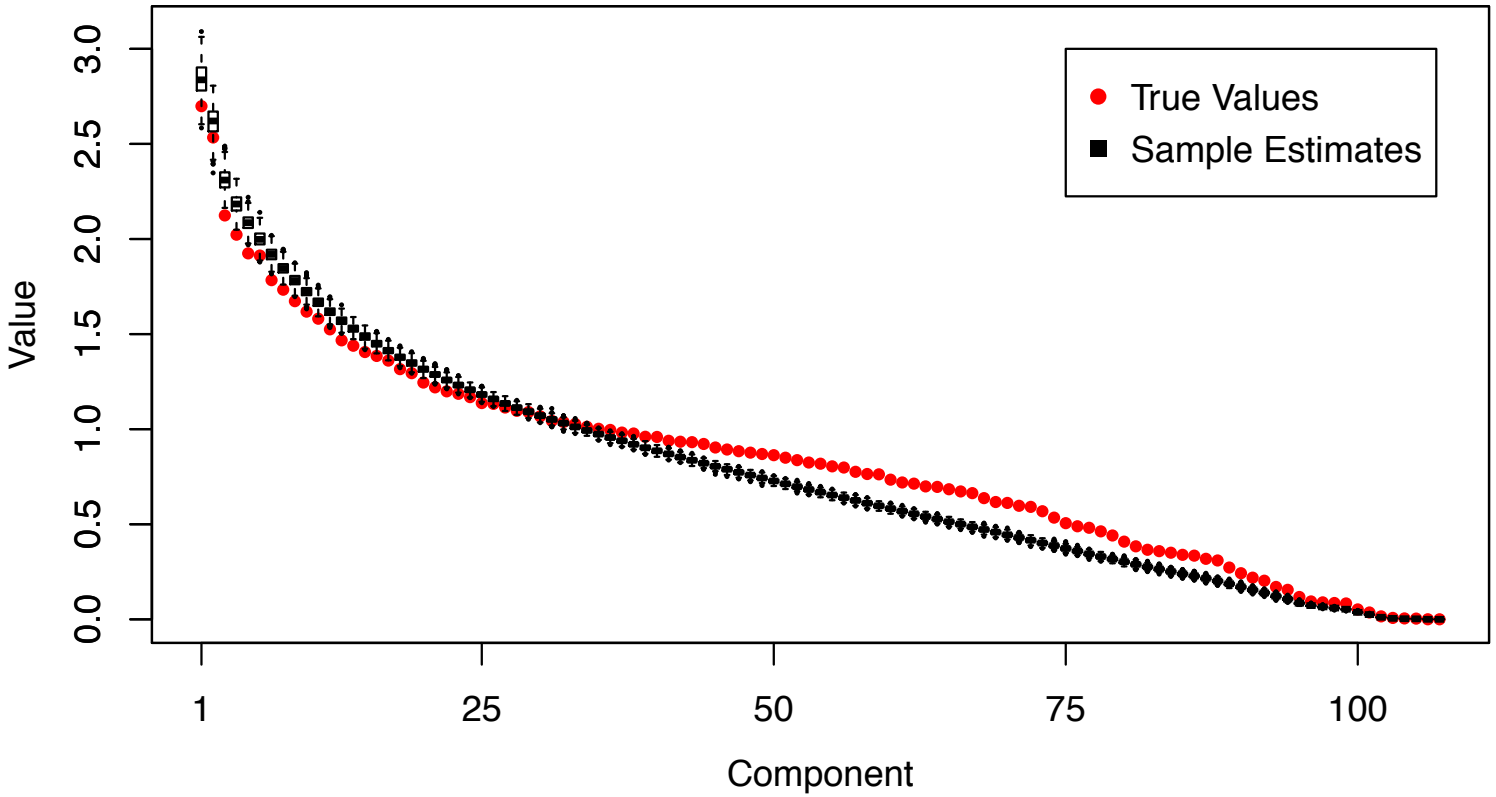


Using Normal, Bernoulli and Binomial RVs, we can derive the overall theoretical false positive rate as 0.3466%.

⟶ this is slightly lower than the empirical false positive rate of 0.5%

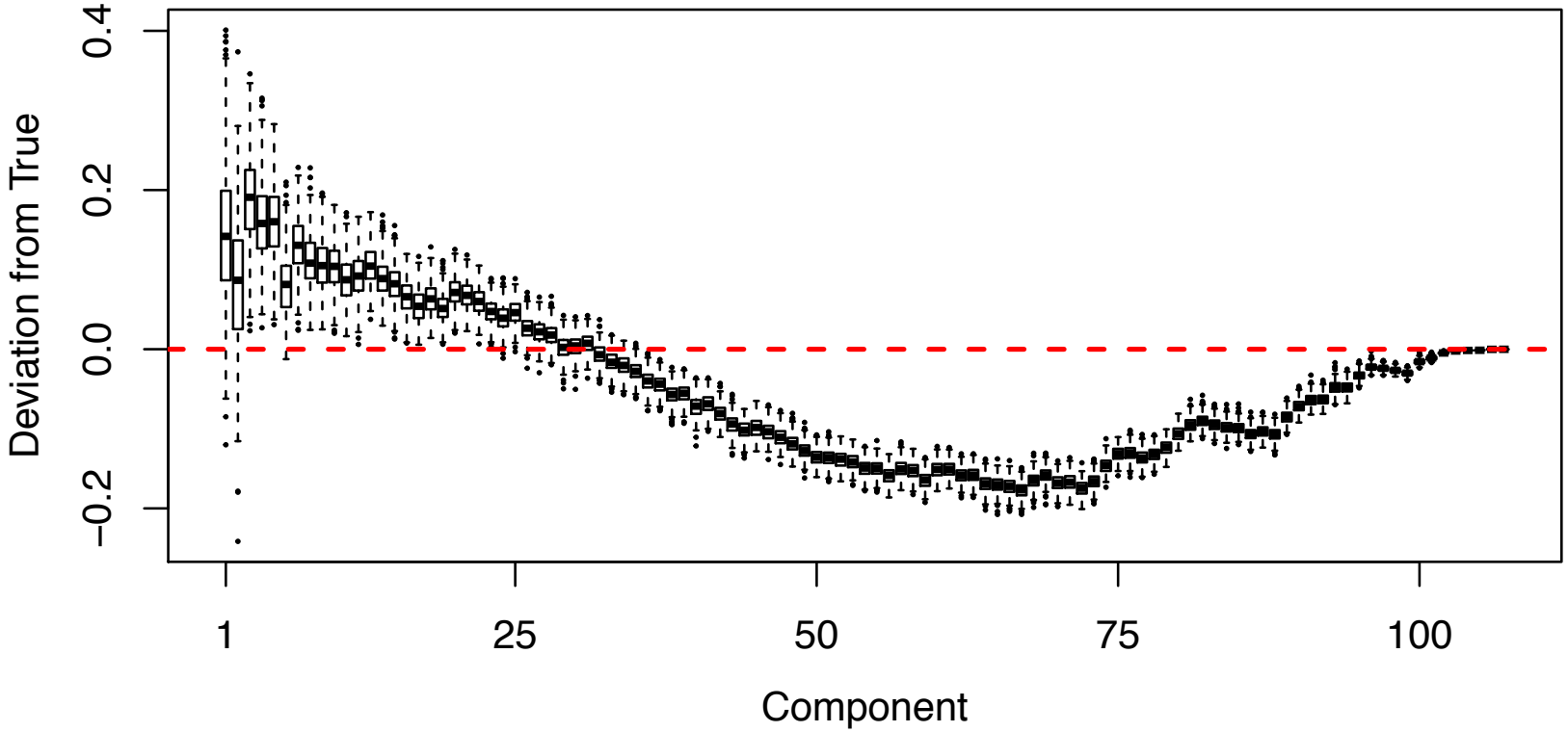
Estimation biases and relation to ensemble size

Illustration of estimation bias of eigenvalues



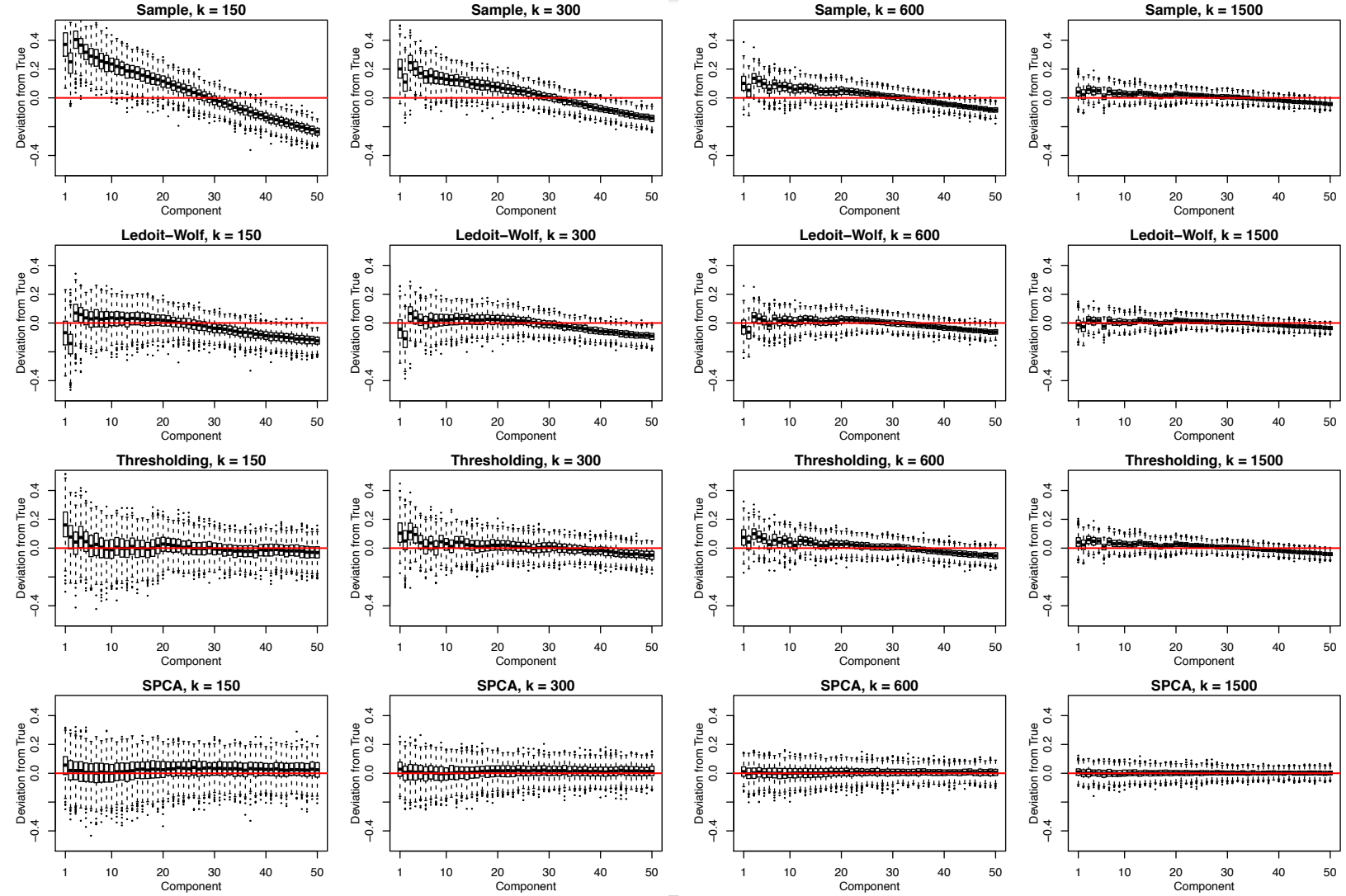
Comparing the true eigenvalues (red) to the estimated eigenvalues from 500 independent simulations from ensembles of size 200 (black).

Illustration of estimation bias of eigenvalues

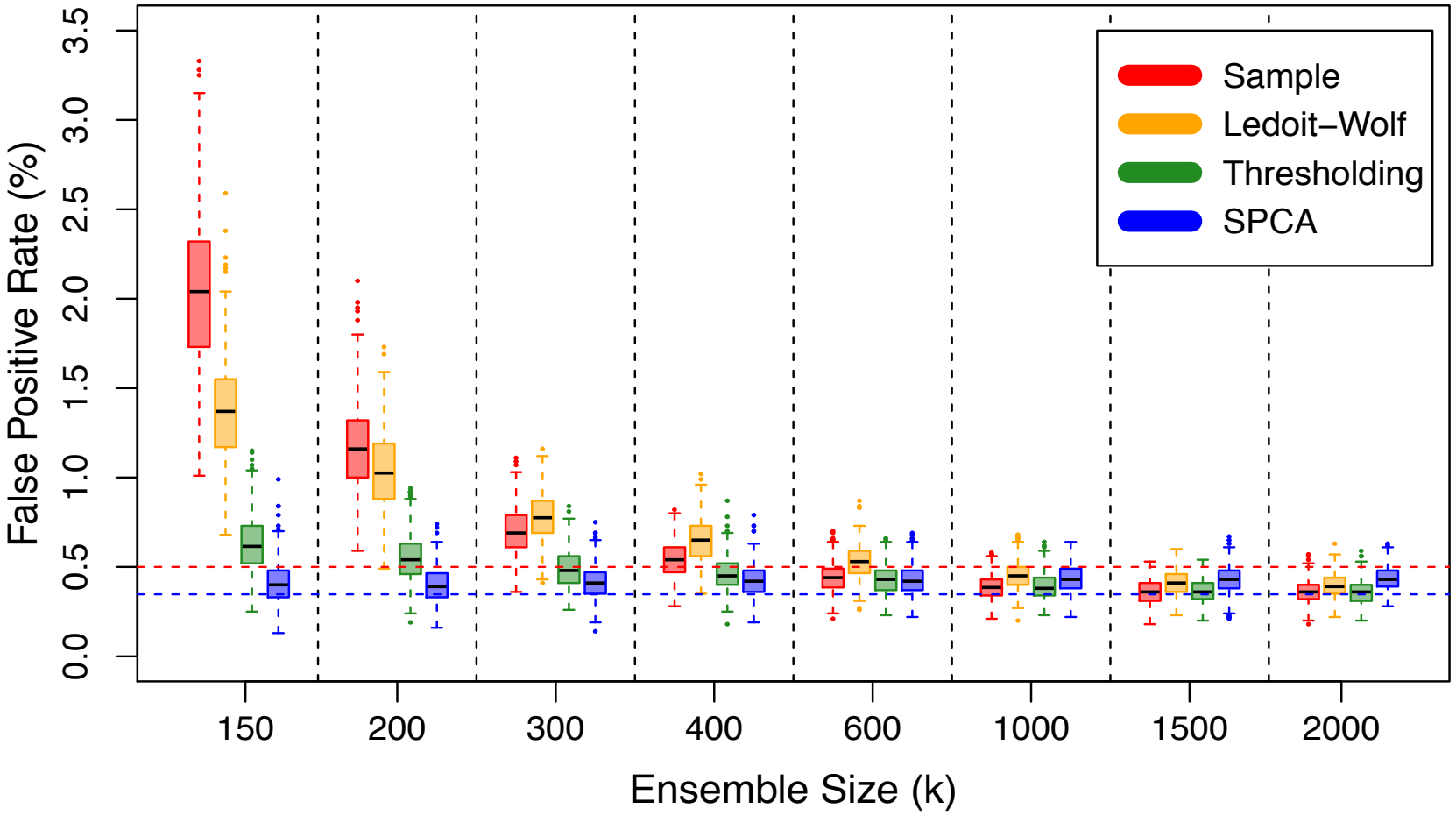


Box plot of the difference between estimated and true eigenvalues from 500 independent simulations from ensembles of size 200. The red dotted line is at zero difference.

Alternative estimators for different ensemble sizes



Estimator and ensemble size effect on false positive rate



When is the PCA-based ECT applicable?

- Assumption of an underlying stochastic process reasonable

- Mean captures the signal

- Large number of output variables

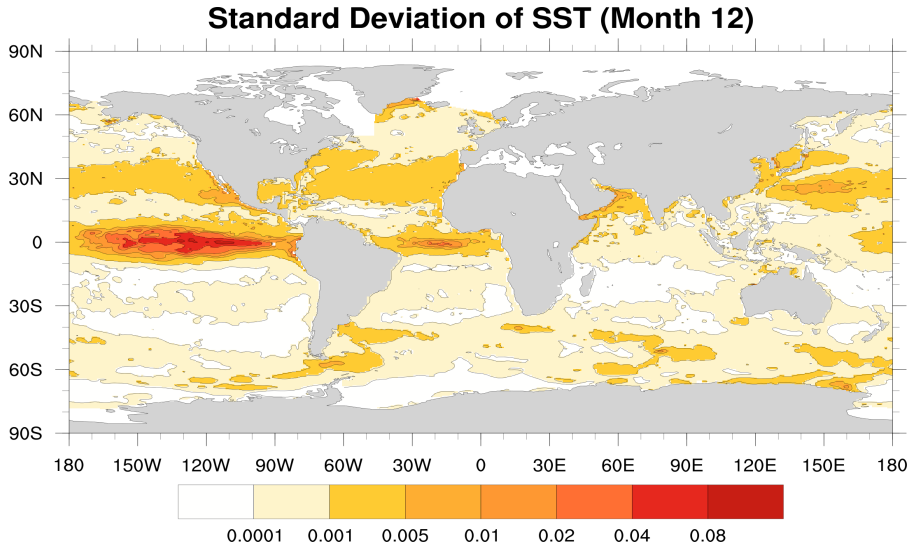
- Feasible to obtain enough ensemble members to estimate true covariance matrix somewhat closely

POP-ECT: for the ocean component

Motivation: evaluating new (more efficient) linear solver for POP

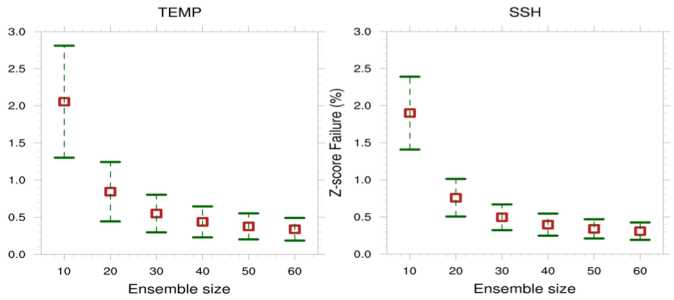
An ensemble approach but conceptually different…

- fewer diagnostic variables (than CAM) – look at individually
- variability is spatially heterogeneous



- smaller ensemble - monthly averages

Evaluate each location in new run against spatially-varying point-wise ensemble variability



Summary and Future Work:

- Using structured hypothesis testing with PCA is useful in the context of numerical models with many output variables.

Summary and Future Work:

- Using structured hypothesis testing with PCA is useful in the context of numerical models with many output variables.

- Accounting for estimation biases in the test procedure is not trivial, but practically of moderate relevance if the ensemble size is large enough. **More on that, and generalization!, in Teo's talk!**

Summary and Future Work:

- Using structured hypothesis testing with PCA is useful in the context of numerical models with many output variables.
- Accounting for estimation biases in the test procedure is not trivial, but practically of moderate relevance if the ensemble size is large enough. **More on that, and generalization in Teo's talk!**
- Overall ensemble consistency testing works surprisingly well for climate models in practice; the much harder open problem is finding the root cause of a failure.

Thanks! Questions? hammerling@mines.edu

References:

A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0). *Geoscientific Model Development*, 2015.

Evaluating statistical consistency in the ocean model component of the Community Earth System Model (pyCECT v2.0).*Geoscientific Model Development*, 2016.

Towards characterizing the variability of statistically consistent Community Earth System Model simulations. *6th Int'l Workshop on Advances in High-Performance Computational Earth Sciences: Applications & Frameworks,* 2016.

Quality Assurance and Error Identification for the Community Earth System Model. *Proceedings of the First Int'l Workshop on Software Correctness for HPC Applications*, 2017.

Nine time steps: ultra-fast consistency testing for the Community Earth System Model (pyCECT v3.0). *Geoscientific Model Development, 2018*.

A statistical investigation of the CESM ensemble consistency testing framework. *NCAR technical note, 2018*.

Making root cause analysis feasible for large code bases: a solution approach for a climate model, *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing: HPDC2019*, 2019.

A PCA-based hypothesis testing framework for large ensembles. *In preparation.*