An ensemble-based consistency test for the Community Earth System Model

Allison Baker

Application Scalability and Performance Group

National Center for Atmospheric Research

Computational & Information Systems Laboratory

CISL

NCAR

International Computing for the Atmospheric Sciences Symposium (ICAS2015)

Many collaborators!

NCAR Earth System Laboratory:

CESM Software Engineering Group

Climate and Global Dynamics Division

NCAR Computational and Information Systems Laboratory:

Application Scalability and Performance Group

Institute for Mathematics Applied to Geosciences

A.H. Baker, D.H. Hammerling, M.N. Levy, H. Xu, J.M. Dennis, B.E. Eaton, J. Edwards, C. Hannay, S. A. Michelson, R. B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein, D. Williamson

+ *D. Milroy*, Computer Science, University of Colorado

CISL Computational & Information Systems Laboratory

NCAR

Software Quality Assurance for CESM

Motivation: To insure that changes during the CESM development life cycle ***do not*** adversely effect the code

- » Code modifications
- » New machine architectures
- » Compiler changes

Main issue: Original data = X

"New" data = $\tilde{X}$

If $X \neq \tilde{X}$ *is the code still "correct"*?

CISL Computational & Information Systems Laboratory

NCAR

Software Quality Assurance for CESM

Motivation: To insure that changes during the CESM development life cycle ***do not*** adversely effect the code

- » Code modifications
- » New machine architectures
- » Compiler changes

Main issue: Original data = X

"New" data = $\tilde{X}$

If $X \neq \tilde{X}$ *is the code still "correct"*?

Does the new data still represent the same climate?

CISL Computational & Information Systems Laboratory

NCAR

Bit-for-Bit?

CESM results are bit-for-bit reproducible if:

The exact *same* code is run,

with *same* parameter settings,

and the *same* initial conditions,

on *same* architecture,

using the *same* compiler,

and the *same* MPI, …

not the case in most applications!

Evaluating the differences…

Question: How to assess whether the difference between

X and $\tilde{X}$ is climate changing ?

Main issue: *There is no clear definition of "climate-changing".*

Previous: Climate scientists compare multiple, long simulations:

computationally intensive, time-consuming, subjective

Need an more objective and easy-to-use methodology!

Evaluating the differences…

New methodology: Leverage climate system's ***natural variability***!

Evaluate new data in the context of an *ensemble* of CESM runs

Evaluating the differences…

New methodology: Leverage climate system's ***natural variability*!**

Evaluate new data in the context of an *ensemble* of CESM runs

- Collection of one-year CESM simulations
- $\mathcal{O}(10^{-14})$ perturbations in initial atmospheric temp.
- "accepted" machine and "accepted" software stack

Creates an "accepted" statistical distribution that can be used to evaluate "new" runs

CESM Ensemble

Composition:

- 151 one-year simulations, annual means
- 1-deg active atm. and land (F-case): **120 variables**

Compare each new variable value to the ensemble distribution:

Issue: variable dependencies

many variables are highly correlated!

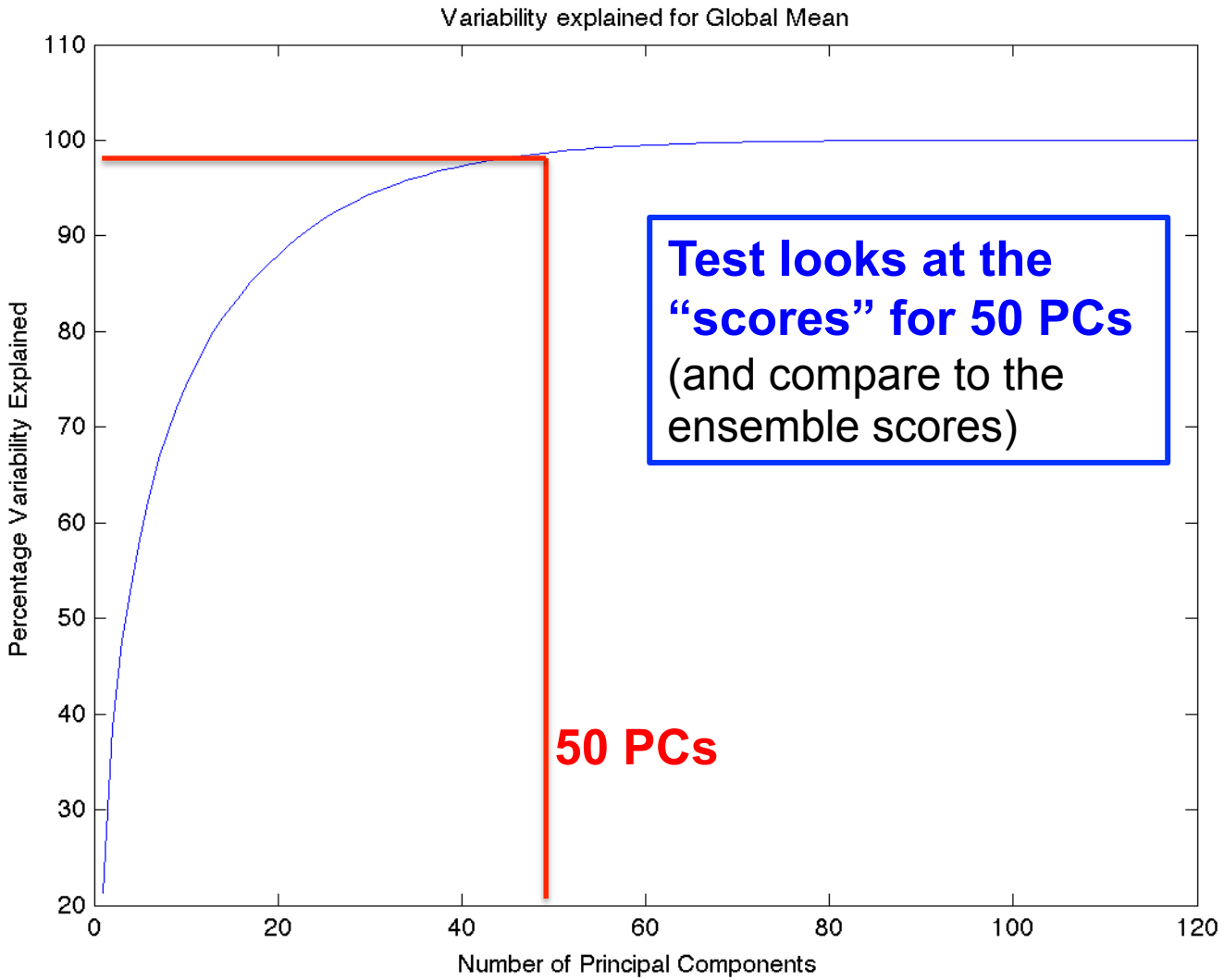
⇒ ***Difficult to make pass/fail choices based on number of variables because of variable dependencies***

=> Principal Component Analysis

Principal Component Analysis (PCA)-based testing

- Rotate (project) data into an *orthogonal* subspace that better represents the variance in the data
- Each component is a linear combination of all 120 variables
- Look only at components that represent the most variance (dimension reduction)
- Can determine a false positive rate

Principal Component Analysis (PCA)-based testing



Test looks at the "scores" for 50 PCs
(and compare to the ensemble scores)

CESM Ensemble Consistency Test

Step 1: *Create an ensemble of CESM runs*

- Use “accepted” machine and “accepted” software stack

Step 2: *Create ensemble summary file*

- Standardize variables
- Determine transformation matrix
- Determine distribution of scores for ensemble

CESM Ensemble Consistency Test

Step 1: *Create an ensemble of CESM runs*

- Use "accepted" machine and "accepted" software stack

Step 2: *Create ensemble summary file*

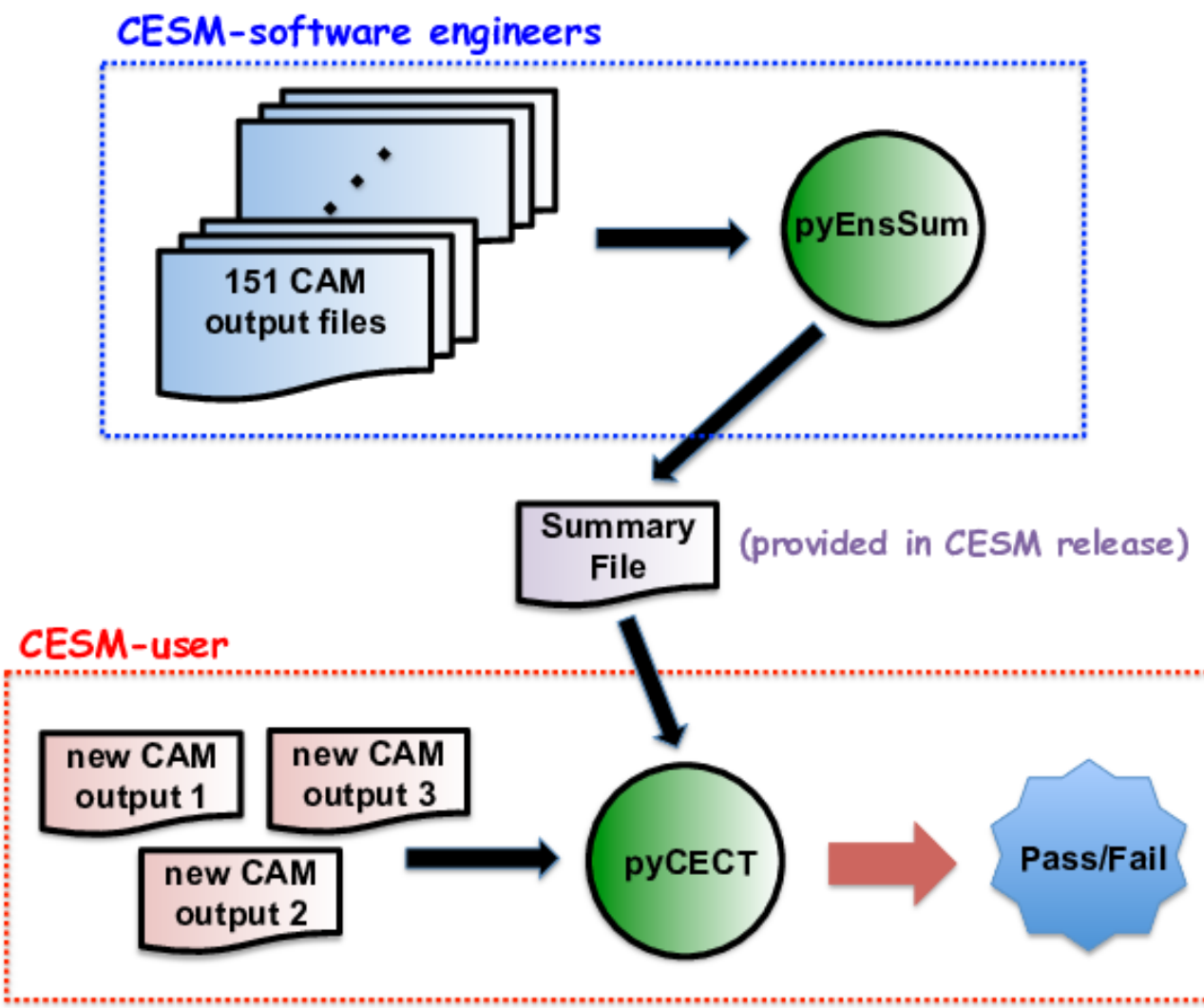
- Standardize variables
- Determine transformation matrix
- Determine distribution of scores for ensemble

Step 3: *Create "new" runs (new platform, code base, …)*

Step 4: *Evaluate new runs*

- Determine new scores (apply transformation matrix)
- Compare new scores to ensemble scores: issue **pass or fail**

CESM Ensemble Consistency Test



CESM Ensemble Consistency Test

Advantages:

- User-friendly (climate-modeling expertise is *not* required)
- Better feedback for model developers
- Flexible accept/reject criteria

Many uses:

- Port-verification (new CESM-supported architectures)
- Heterogeneous computing platforms
- Exploration of new algorithms, solvers, compiler options, …
- Evaluation of data compression on CESM data

Does it work?

Initial Experiments:

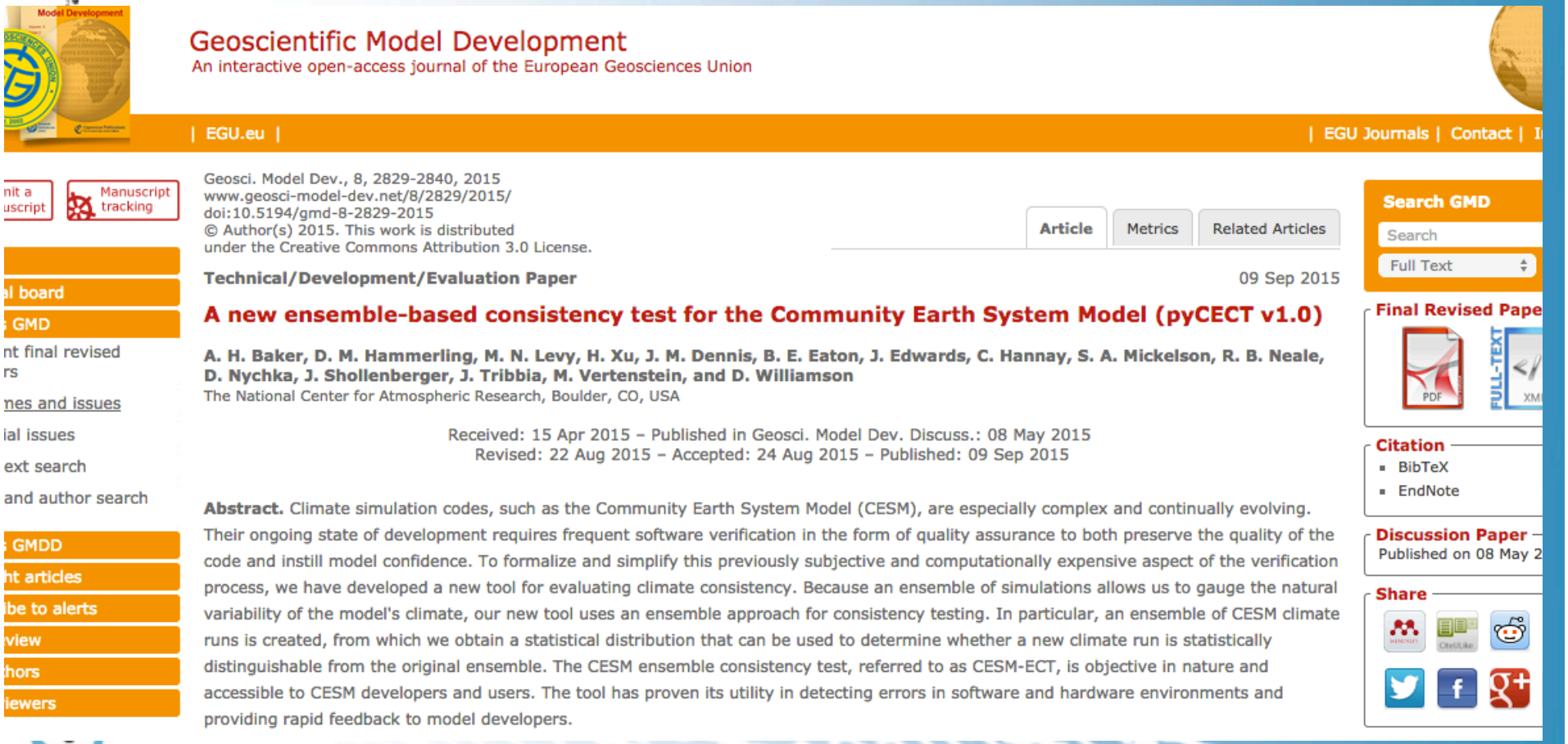
- **Modifications not expected to be climate-changing**
  - 5 of 5 compiler and threading modifications *pass*
- **Modifications expected to be climate-changing**
  - 10 of 11 CAM parameter modifications *fail*
- **CESM-supported machines as modifications**
  - Some borderline failures - *Currently investigating*

Practical applications

- **Error in cloud generator code only manifested on big endian machine**
  - ❖ Decisive failures on big endian machine
- **Errors in new version of Community Ice Code**
  - ❖ Not detected in standalone component testing

Test name	CESM-ECT Results	Number of PCs failing at least 2 runs
CICE4-INTEL	PASS	1
CICE4-GNU	PASS	0
CICE4-PGI	PASS	0
CICE5-INTEL	FAIL	19
CICE5-GNU	FAIL	20
CICE5-PGI	FAIL	19

Paper and code available



Geoscientific Model Development
An interactive open-access journal of the European Geosciences Union

| EGU.eu | | EGU Journals | Contact |

Manuscript tracking

Geosci. Model Dev., 8, 2829-2840, 2015
www.geosci-model-dev.net/8/2829/2015/
doi:10.5194/gmd-8-2829-2015
© Author(s) 2015. This work is distributed
under the Creative Commons Attribution 3.0 License.

Article | Metrics | Related Articles

Search GMD
Search
Full Text

Technical/Development/Evaluation Paper

09 Sep 2015

A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0)

A. H. Baker, D. M. Hammerling, M. N. Levy, H. Xu, J. M. Dennis, B. E. Eaton, J. Edwards, C. Hannay, S. A. Mickelson, R. B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein, and D. Williamson
The National Center for Atmospheric Research, Boulder, CO, USA

Received: 15 Apr 2015 – Published in Geosci. Model Dev. Discuss.: 08 May 2015
Revised: 22 Aug 2015 – Accepted: 24 Aug 2015 – Published: 09 Sep 2015

Abstract. Climate simulation codes, such as the Community Earth System Model (CESM), are especially complex and continually evolving. Their ongoing state of development requires frequent software verification in the form of quality assurance to both preserve the quality of the code and instill model confidence. To formalize and simplify this previously subjective and computationally expensive aspect of the verification process, we have developed a new tool for evaluating climate consistency. Because an ensemble of simulations allows us to gauge the natural variability of the model's climate, our new tool uses an ensemble approach for consistency testing. In particular, an ensemble of CESM climate runs is created, from which we obtain a statistical distribution that can be used to determine whether a new climate run is statistically distinguishable from the original ensemble. The CESM ensemble consistency test, referred to as CESM-ECT, is objective in nature and accessible to CESM developers and users. The tool has proven its utility in detecting errors in software and hardware environments and providing rapid feedback to model developers.

Final Revised Pape
PDF | FULL-TEXT

Citation
- BibTeX
- EndNote

Discussion Paper
Published on 08 May 2

Share

https://github.com/NCAR-CISL-ASAP/PyCECT/releases

NCAR

How to create the ensemble?

Effectiveness of CESM-ECT method relies heavily on the "accepted" ensemble composition

- *size 151, 1-year, Yellowstone machine, Intel compiler*
- *perturbing the initial condition (IC) for atmospheric temp.*

Does the original CESM-ECT ensemble represent the variability of a consistent climate?

- *How well do IC perturbation capture "legitimate" differences?*
- *Is the current distribution sufficient to capture compiler and code changes?*

More extensive testing…

Testing Ensemble Composition

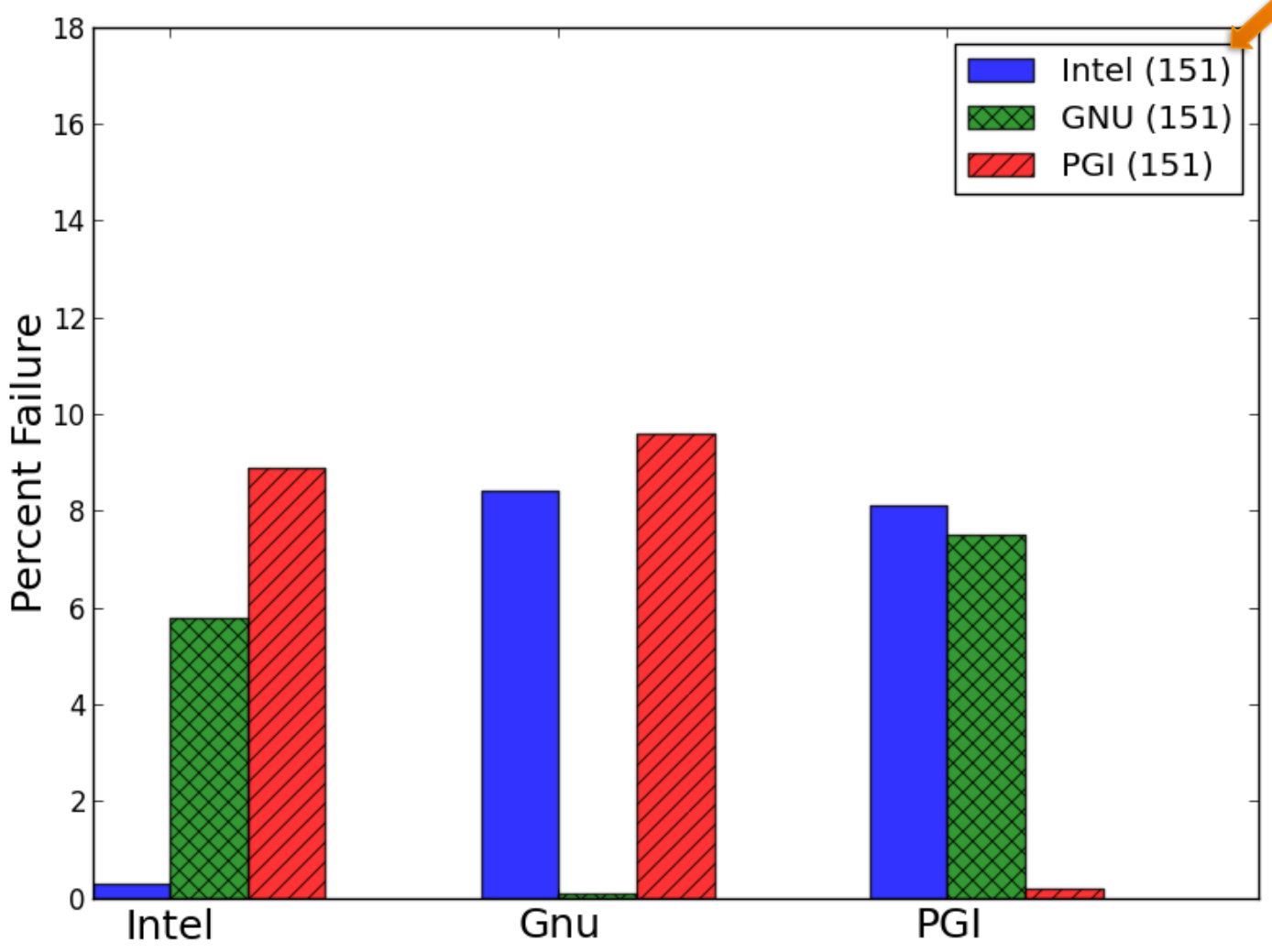
Do we need different compilers represented in the ensemble?

More tests:

1. Repeat IC experiments using all 3 CESM-supported Yellowstone compilers (intel, pgi, gnu) – 181 members
2. Same perturbations in initial conditions
3. 120 variables

Ensemble composition

Is the current ensemble distribution sufficient to capture compiler changes?



Initial Condition effect?

Random draws

-Exclude 30 at random from each 181-member set

-Test the excluded set

Failure %

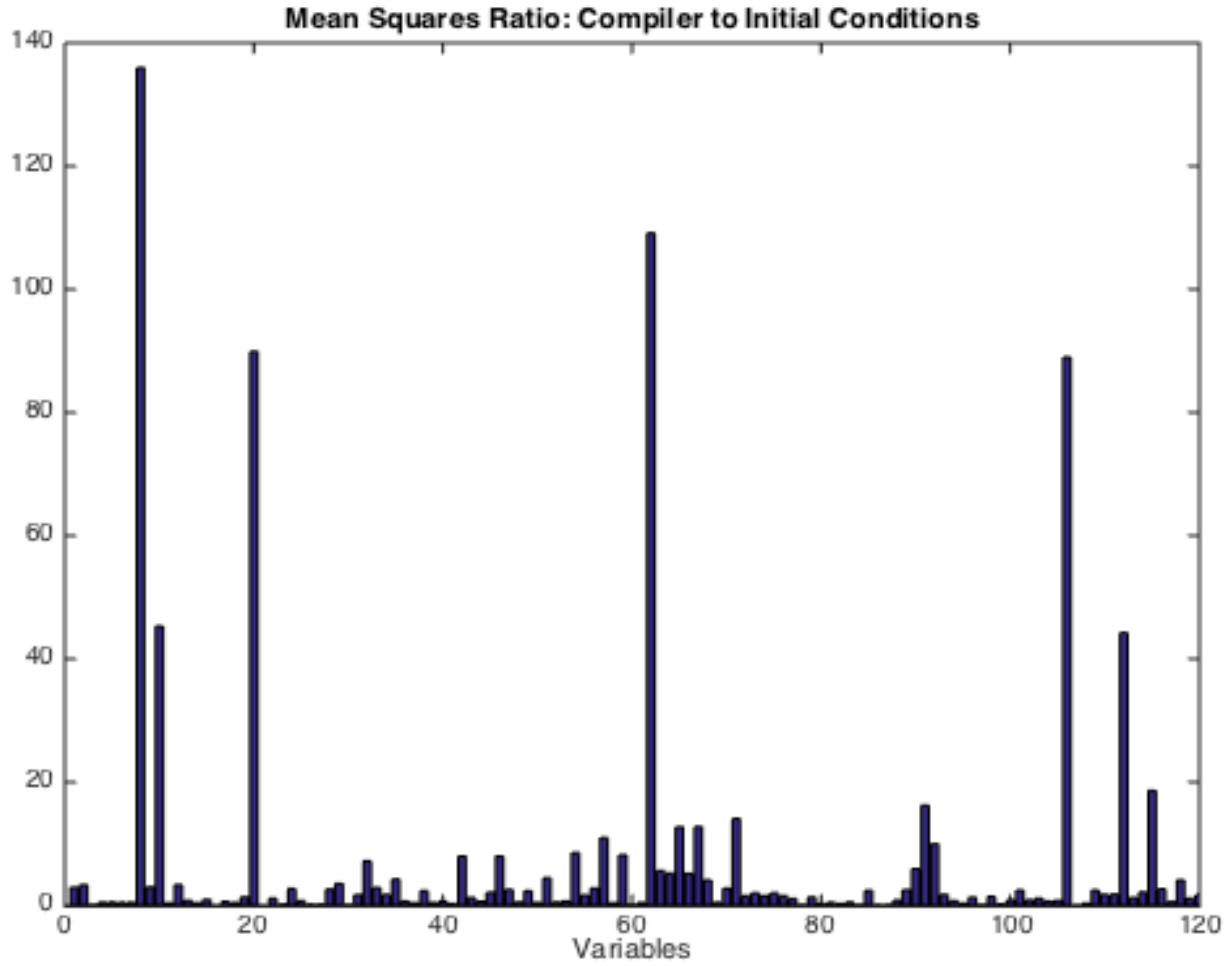
Exp.	Intel-1 Ens	Intel-2 Ens.	Intel-3 Ens.	Ggu-1 Ens.	Gnu-2 Ens.	Gnu-3 Ens.	Pgi-1 Ens.	Pgi-2 Ens.	Pgi-3 Ens.
Rand-Intel-1	**8.1**	0.0	0.6	9.5	3.5	3.9	4.5	2.3	6.0
Rand-Intel-2	0.1	**4.8**	1.9	4.1	11.0	5.8	7.7	2.4	7.8
Rand-Intel-3	0.3	0.2	**8.3**	8.9	8.7	8.7	5.2	4.4	12.7

Don't want pass/fail dependent on which random sample from ensemble …

NCAR

Compiler vs. IC effects

Two-way analysis of variance (ANOVA):



CISL Computational & Information Systems Laboratory

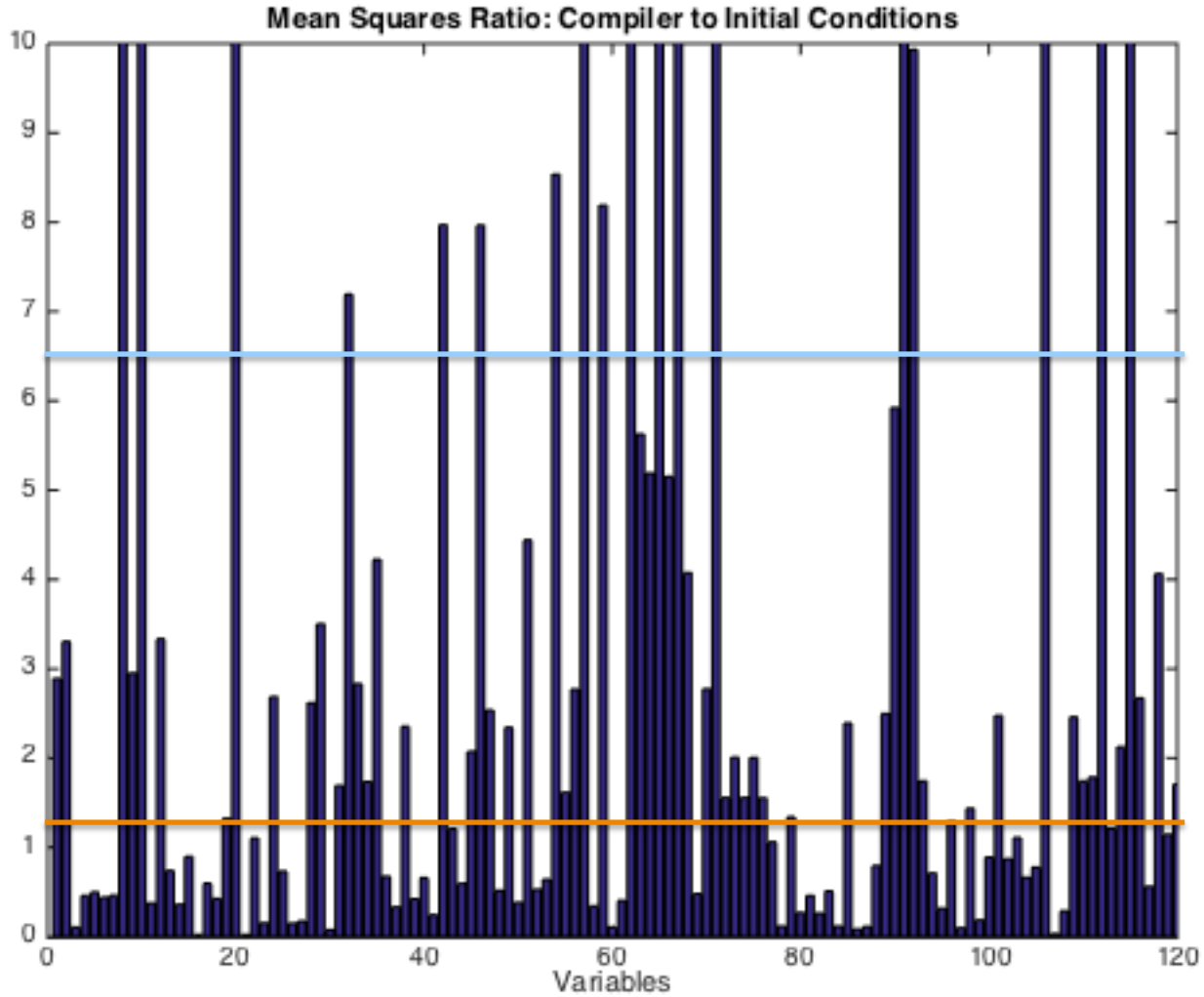
NCAR

Compiler vs. IC effects

Two-way analysis of variance (ANOVA):

Mean=6.6

Median=1.3



Compiler has more influence….

Refine Ensemble

- ***Compiler aggregate ensembles (size 453)***

Experiments	Intel-Gnu-PGI Rand-1	Intel-Gnu-PGI Rand-2	Intel-Gnu-PGI Rand-3
Rand-Intel-1	0.8	0.0	0.5
Rand-Intel-2	0.2	1.8	0.4
Rand-Intel-3	0.8	0.5	1.9

- ***Low failure rates***

Simple code changes?

Is the current ensemble distribution sufficient to capture reasonable code changes to CAM?

(e.g., mathematically identical and "small")

Example 1 ***(Combine)***

Original:

```
omega_p(i,j,1) = vgrad_p(i,j,1)/p(i,j,1)
omega_p(i,j,1) = omega_p(i,j,1) - 0.5d0/p(i,j,1)*divdp(i,j,1)
```

Modified:

```
omega_p(i,j,1) = (vgrad_p(i,j,1) - 0.5d0*divdp(i,j,1))/p(i,j,1)
```

Simple code changes?

Example 2 ***(Expand)***

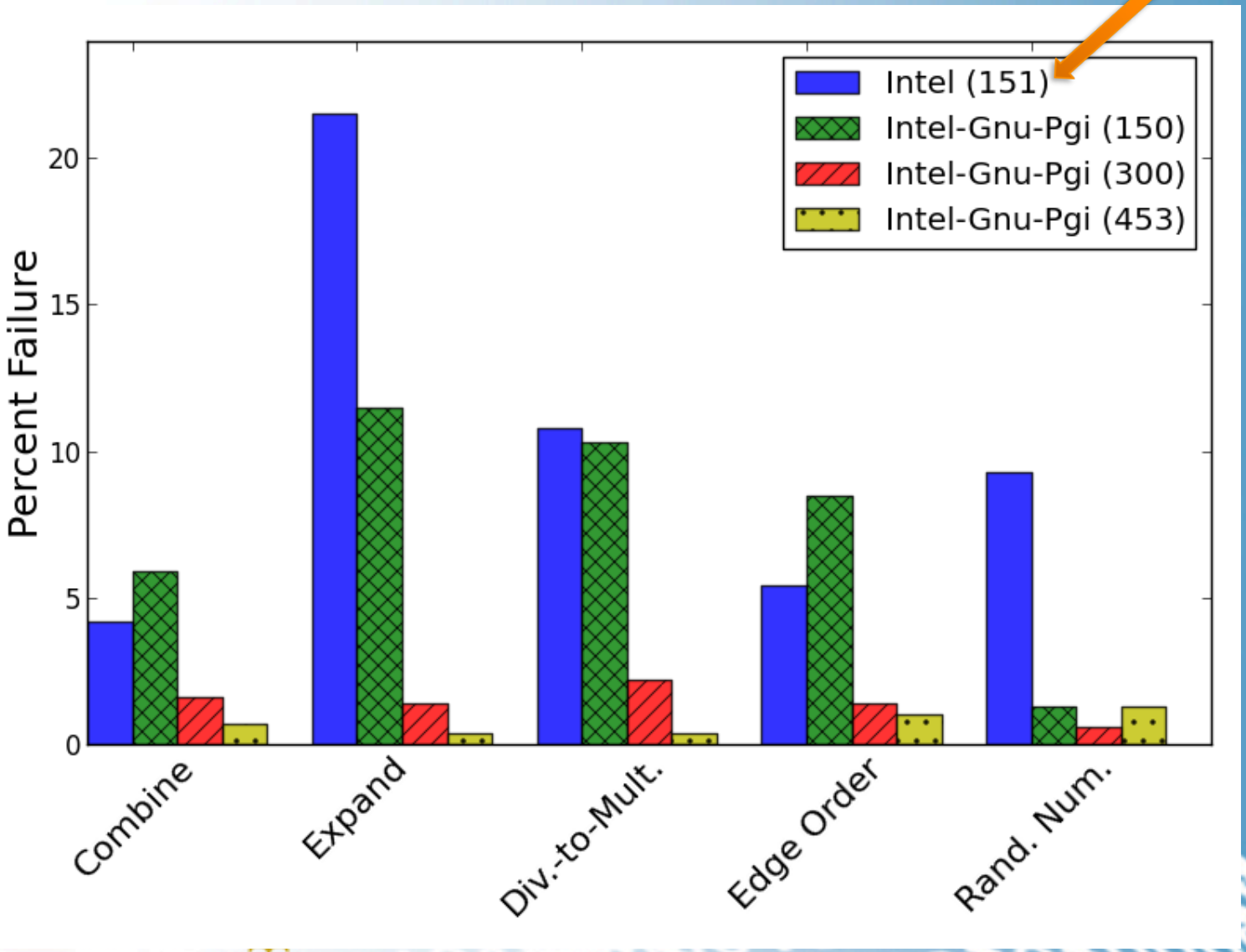
Original:

```
phii(i,j,nlev)  = Rgas*T_v(i,j,nlev)*hkl
phi(i,j,nlev) = phis(i,j) + Rgas*T_v(i,j,nlev)*hkk
```

Modified:

```
phii(i,j,nlev) = T_v(i,j,nlev)*hkl
phii(i,j,nlev) = Rgas*phii(i,j,nlev)
tt_real = T_v(i,j,nlev)*hkk
phi(i,j,nlev) = phis(i,j) + Rgas*tt_real
```

Simple code changes?



Next Steps

1) Investigate ensemble size

(stability of PC calculations)

2) Length of ensemble runs

(shorter?)

Next Steps

1) *Investigate ensemble size*
   (stability of PC calculations)

2) *Length of ensemble runs*
   (shorter?)

3) ***Fine-grained testing capability for failures***
   (to identify groups of variables that cause failure)

Next Steps

1) *Investigate ensemble size*
   (stability of PC calculations)

2) *Length of ensemble runs*
   (shorter?)

3) *Fine-grained testing capability for failures*
   (to identify groups of variables that cause failure)

4) ***Evaluate spatial patterns in addition to global***
   (e.g. regional features, boundaries ocean/land, spatial structure)

5) ***Evaluate spatial relationships between variables***
   (cross-covariance studies)

Thanks!