

Sampler Tutorial Overview

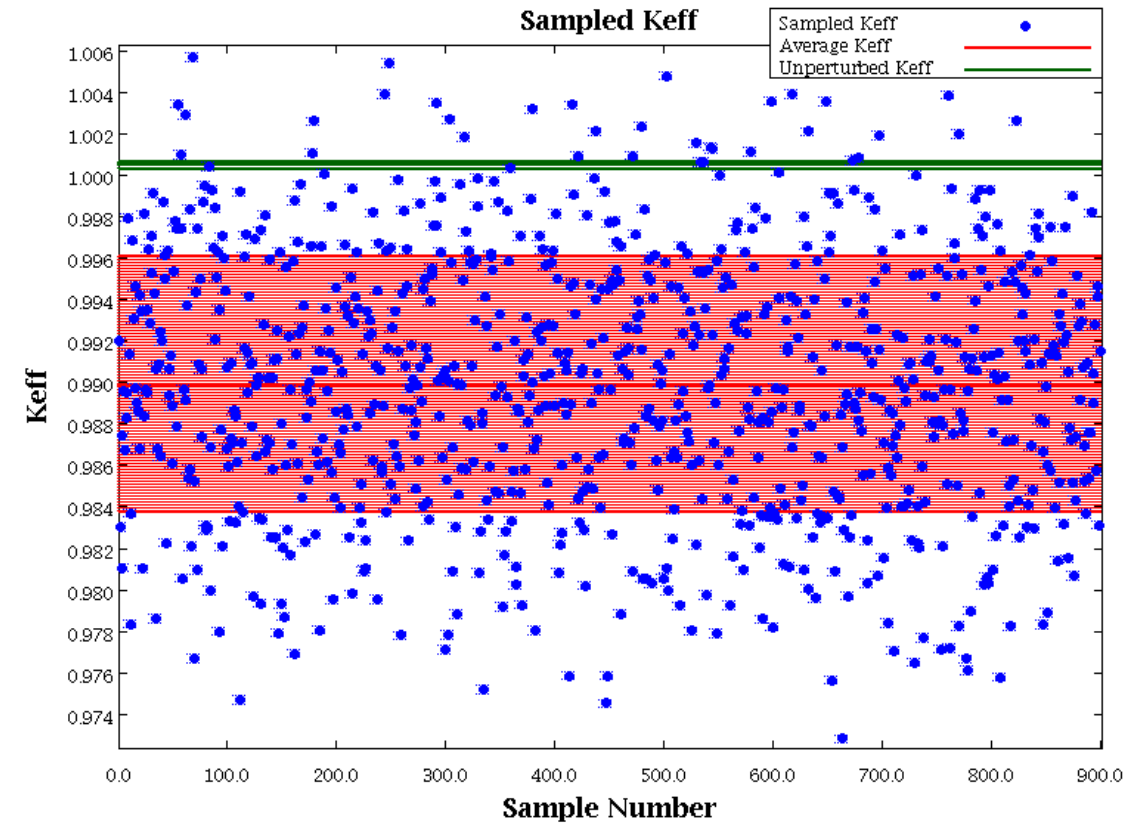


Sampler: A Module for Statistical Uncertainty Analysis with SCALE Sequences

Calculates uncertainty in *any* result of a SCALE computation* due to two basic types of uncertainty

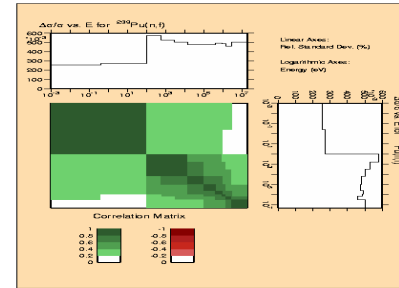
- nuclear data uncertainty
 - cross sections
 - fission yield data
 - decay data
- input parameter uncertainty
 - geometry
 - composition
 - *and anything else that appears in an input file*

*multi-group only in SCALE 6.2



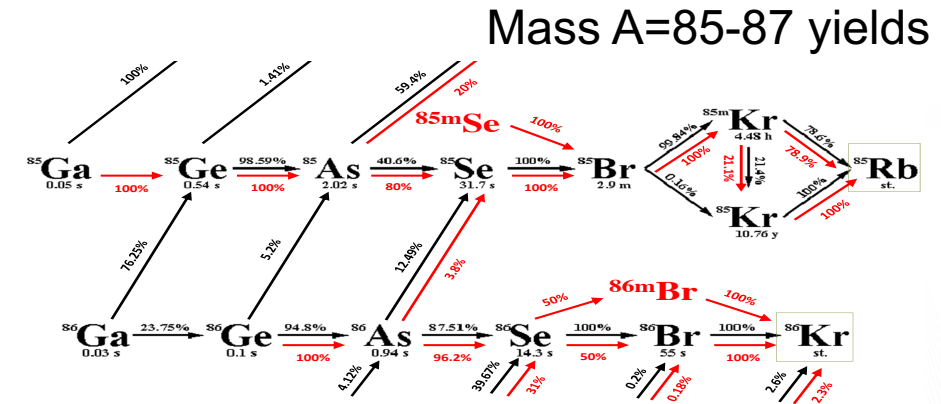
Sources of Sampler Nuclear Data Uncertainty

- Cross section covariances:
 ENDF-VII.1 supplemented by other sources
 (SCALE cov. library)



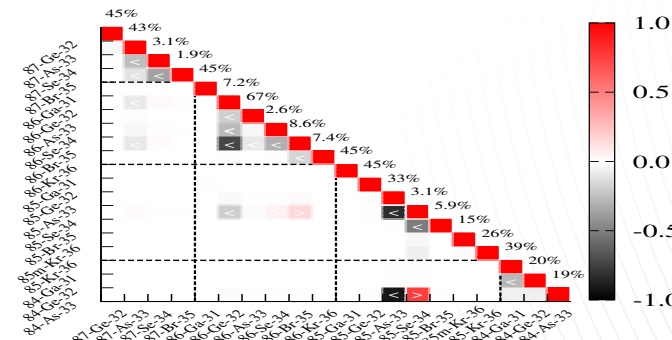
Pu-239 fission covariance

- Fission product yield:
 Standard deviations from ENDF/B-VII; correlations generated by combining independent and cumulative yields



Mass A=85-87 yields

- Decay data:
 ENDF-VII.1 modified to include branching correlations due to constraint that branch sum=1.0

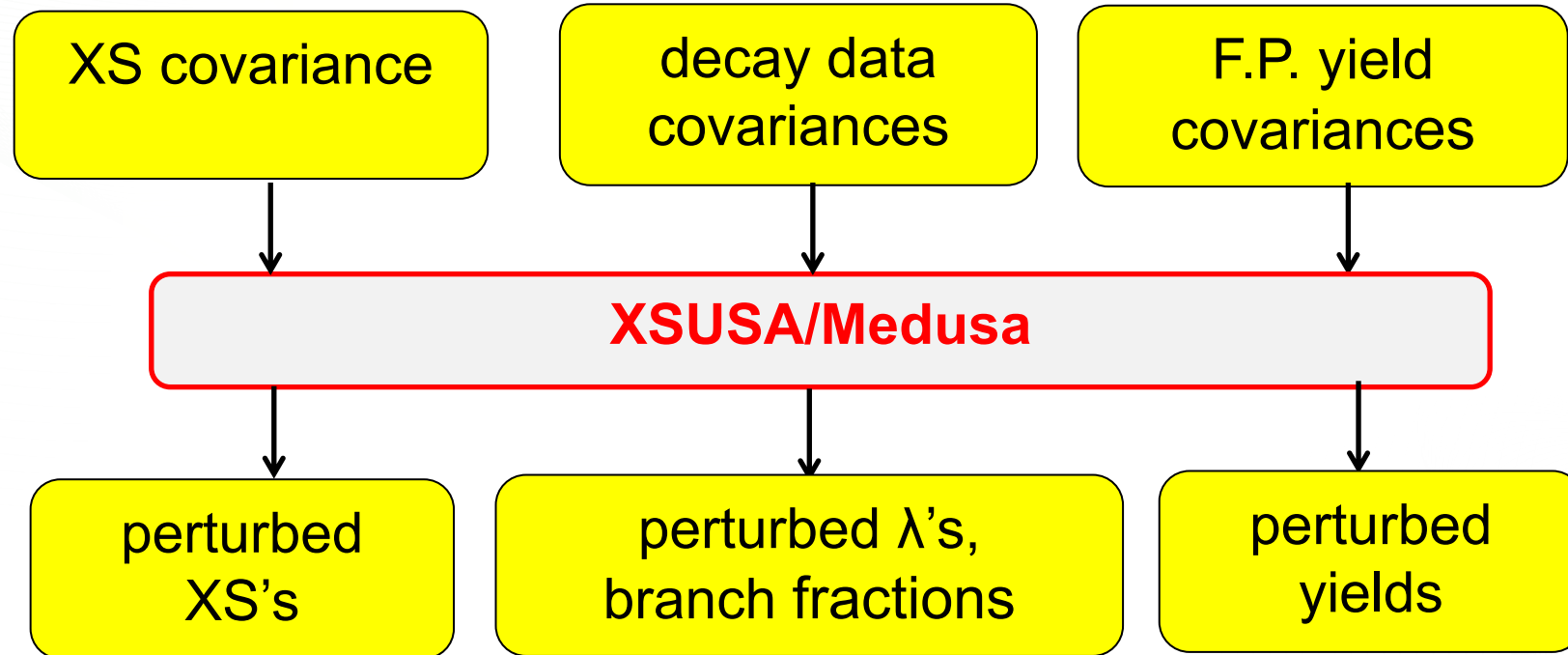


Yield covariance

How does Sampler work?

- SCALE provides covariance data for neutron cross sections, fission product yields and decay
- For a given sample,
 - the nuclear data is perturbed based on a Gaussian probability distribution characterized by the covariance data
 - A set of perturbed XS libraries (XS, CE XS, FP yields, decay) are created for each random sampling
 - The SCALE input (1 or more) is run with the perturbed libraries
 - The user requested output data and perturbed archive files (e.g. t16 files) are collected
- After the calculation, SAMPLER can post-process SCALE output into means, standard deviations, and correlation coefficients
- SAMPLER can also perturb parameters in the input files based on a user specified probability distribution

Generation of Perturbed Nuclear Data Samples



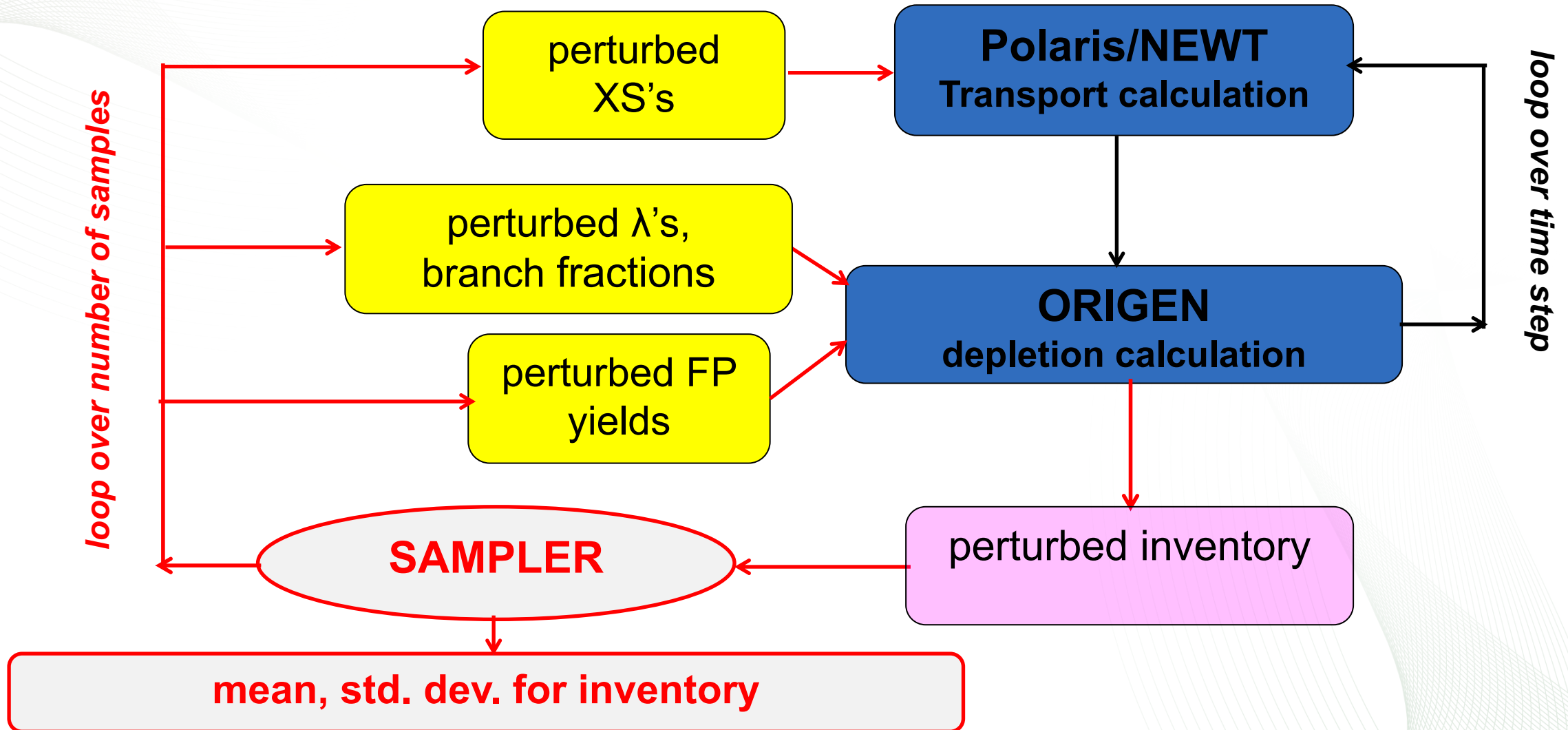
ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method • Nashville, TN • April 19-23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

UNCERTAINTY AND SENSITIVITY ANALYSIS IN CRITICALITY CALCULATIONS WITH PERTURBATION THEORY AND SAMPLING

Friederike Bostelmann, Frank-Peter Weiß, Alexander Aures, Kiril Velkov,
Winfried Zwermann
Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH
Forschungszentrum, Garching, Germany, 85748
Winfried.Zwermann@grs.de, Frank-Peter.Weiss@grs.de, Alexander.Aures@grs.de,
Friederike.Bostelmann@grs.de, Kiril.Velkov@grs.de

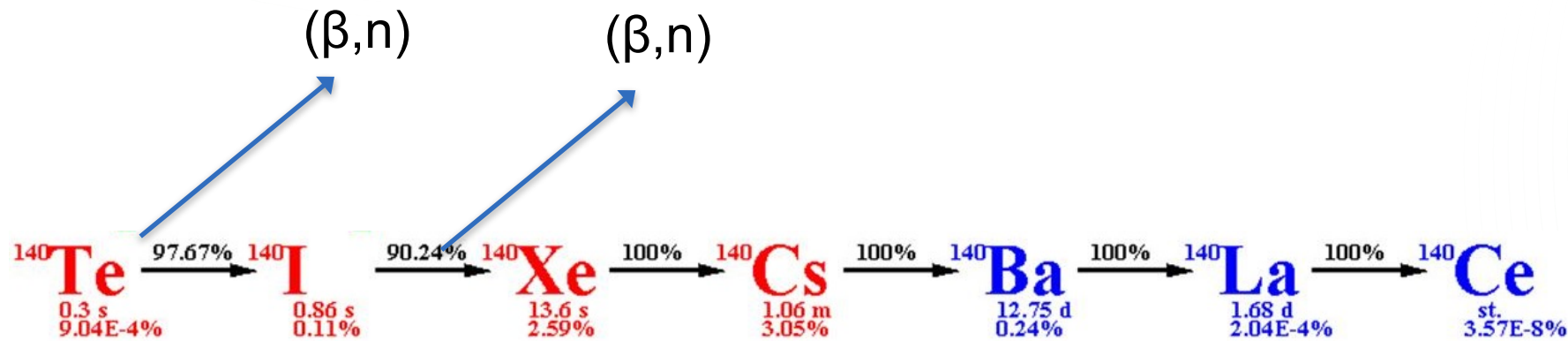
Bradley T. Rearden, Matthew A. Jessee, Mark L. Williams, Dorothea Wiarda,
William A. Wieselquist
Oak Ridge National Laboratory
P.O. Box 2008, Oak Ridge, TN 37831-6354, USA
reardenb@ornl.gov, jesseema@ornl.gov, williamsml@ornl.gov, wiardada@ornl.gov,
wieselquistwa@ornl.gov

Extending Nuclear Data Uncertainty to Depletion Data



Uncertainties In Fission Product Chains

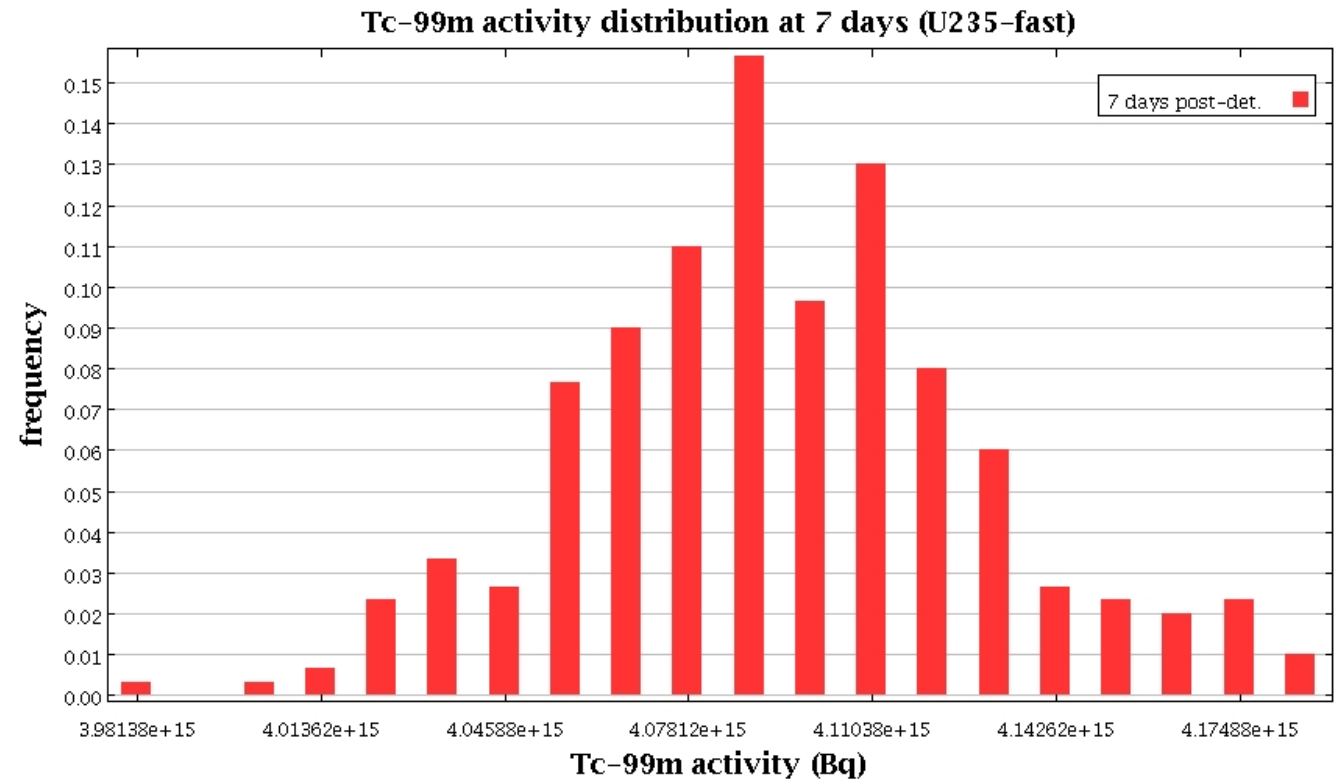
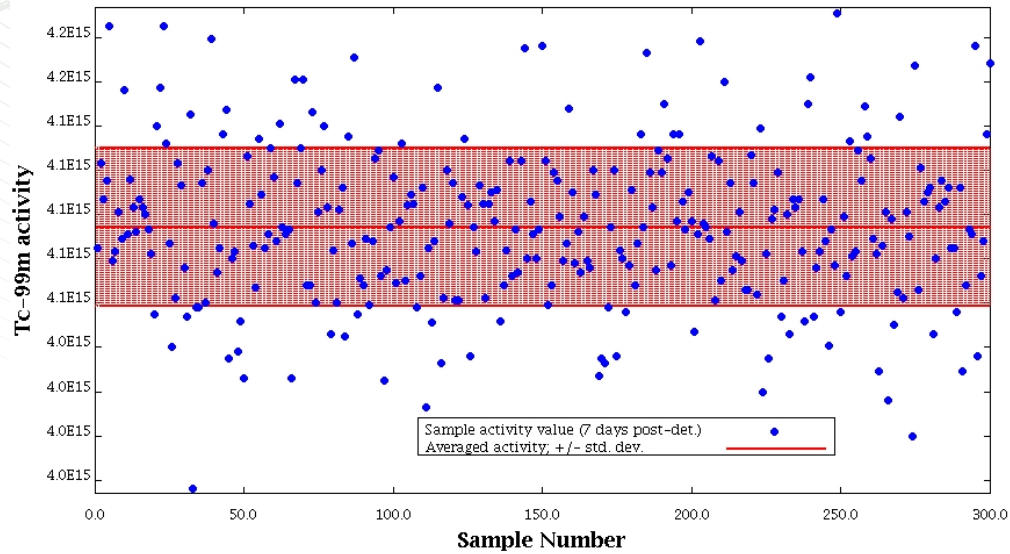
- Independent FP yields are mainly from model calculations...
→ have large uncertainties
- Cumulative FP yields based largely on measurements...
→ have smaller uncertainties
- Beta-delayed neutron branching ratios
→ usually have large uncertainties



half-life

integral fission yield

Example Impact of FP Yield and Decay Data Uncertainties For U235 Fission Burst



Nuclear data perturbations are easy

```
read parameters
  n_samples      = 100
  perturb_xs     = yes
  perturb_decay  = yes
  perturb_yields = yes
end parameters
```

```
=%sampler
read parameters
  n_samples      = 100
  perturb_xs     = yes
  perturb_decay  = yes
  perturb_yields = yes
end parameters

read case[c1]
sequence=t-depl parm=(bonami,addnux=0)
pincell model
xn238v7
read composition
uo2      10 0.95 900 92235 3.6 92238 96.4 end
zr-90    20 1 600 end
h2o      30 den=0.75 0.9991 540 end
end composition
read celldata
  latticecell squarepitch pitch=1.2600 30 fuelr=0.4095 10 cladr=0.4750 20 end
end celldata
read depletion
  10
end depletion
read burndata
  power=25 burn=60 nlib=2 down=30 end
end burndata
read model
read materials
...
```

Perturbation of Compositions and Geometry

- Can be used to quantify uncertainties, or to calculate correlation coefficients
- User selects appropriate distribution and parameters for sampling composition and geometry inputs
 - Available distributions: uniform, normal (*can be truncated*), and beta
- Simple mathematical expressions can also be used to calculate perturbed inputs
- Perturbations applied to specified cases allowing identical realizations for shared characteristics
- Case input can be included in Sampler input or modified via the SCALE Input Retrieval Engine (SIREN)

Imbedded Input (aka Placeholders)

SCALE Standard Composition

```
read composition
uranium 1 den=18.742 1 300
          92235 93.7112
          92238 5.2686
          92234 1.0202 end
end composition
```

Placeholder Variables

```
read composition
uranium 1 den=18.742 1 300
          92235 #{u235_wo}
          92238 #{u238_wo}
          92234 #{u234_wo} end
end composition
```

Variable Examples

```
read variable[u234_wo]  
  distribution=uniform  
  value=1.0202  
  min=1  
  max=1.0404  
end variable
```

```
read variable[u235_wo]  
  distribution=normal  
  value=93.7112  
  stddev=0.05  
  min=93.5  
  max=93.9224  
end variable
```

```
read variable[u238_wo]  
  distribution=expression  
  expression="100.0 - u235_wo - u234_wo"  
end variable
```


Post-processing

- **Creates response tables as requested**
 - Average values, correlations, covariances, etc.
- **Correlations created for multiple cases** (e.g. critical experimental correlations)
- **PTP plot files**
 - Running averages, histograms, scatter plots

Average and std. deviation table for burnup-independent variables	
godiva:keff	godiva:vars.radius
1.0019e+00 +/- 2.3944e-03 (0.24 %)	8.7563e+00 +/- 1.8309e-02 (0.21 %)

Summary

- Sampler is a wrapper for uncertainty quantification in any (multi-group) SCALE calculation
- Leading-edge nuclear data uncertainty
 - Fission product yield uncertainty
 - Decay data uncertainty
 - Multi-group cross section uncertainty
- Convenient input parameter perturbations
 - assign distributions to variables
 - use variables in placeholders `#{myvariable}`
 - or assign to distributions to input without copy/paste/placeholders via SIREN expressions

Sampler Tutorial

Decay uncertainty propagation



Uncertainty in calculations of Rn-222 activity

Problem

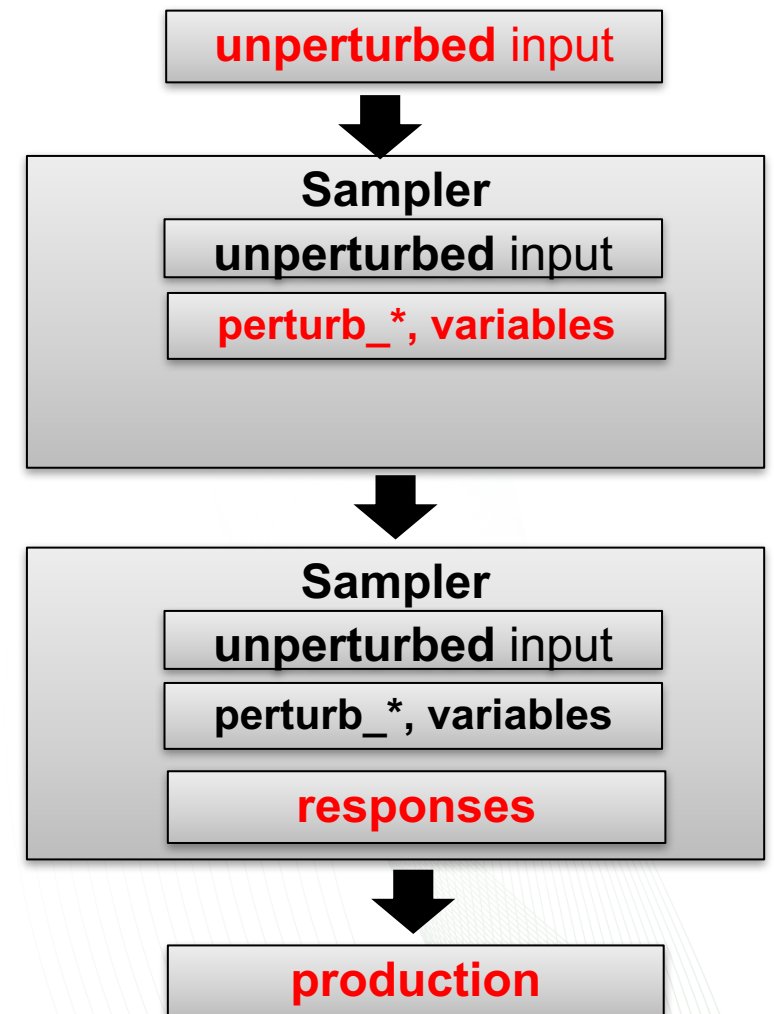
Decay 1 mol of U-238 for 1 billion years

Goals

- Determine uncertainty in Radon-222 (^{222}Rn) activity
- Gain familiarity with
 - the Sampler process
 - Sampler directories and results files
- Discuss limits of decay uncertainty calculation

The Sampler Process

1. Create the "unperturbed" SCALE input
 - be certain that it **works!**
 - make sure it produces the outputs you want (f71, plt, etc.)
2. Wrap it in Sampler
 - use `n_samples=2, force_run=yes, run_cases=no`
 - verify you have the desired `perturb_*=yes`
 - verify variables are sampled correctly
3. Verify responses
 - use `n_samples=2, force_run=no, run_cases=yes`
 - NOTE: **greps** almost always take a few iterations
4. Start "production" run
 - `n_samples=100` (or more)!

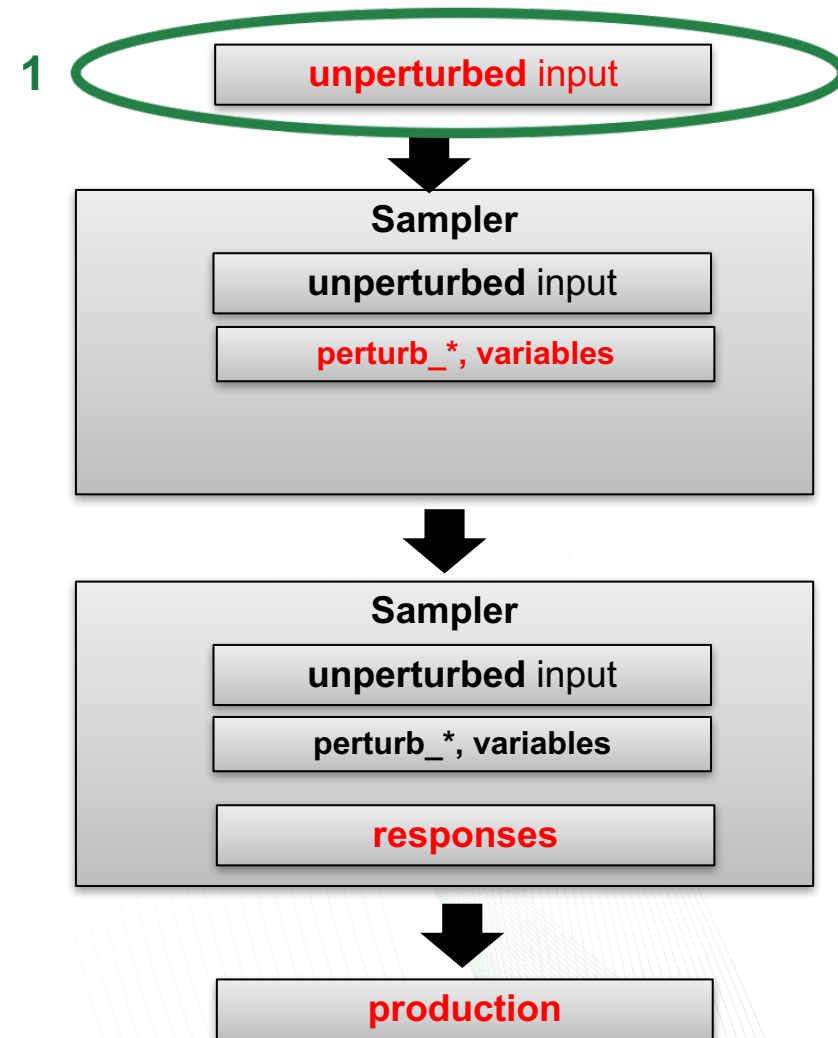


The Sampler Process: Step 1

Create the "unperturbed" SCALE input

- **be certain that it works!**
- make sure it produces the outputs you want (f71, plt, etc.)

```
'rn222.inp decays 1 mol u238 for 1e9 years
' produces output: f71 (moles) and plt (curies)
=origen
case{
  title="U238 decay chain"
  lib{ file="end7dec" }
  time{ units=years t=[3L 1e1 1e9] }
  mat{ iso=[u238=1.0] }
  print{ nuc{ units=[moles curies] } }
  save{ file="ft71f001"
        steps=[0 1 2 3 4 5]
      }
}
end
=opus
units=curies
end
```



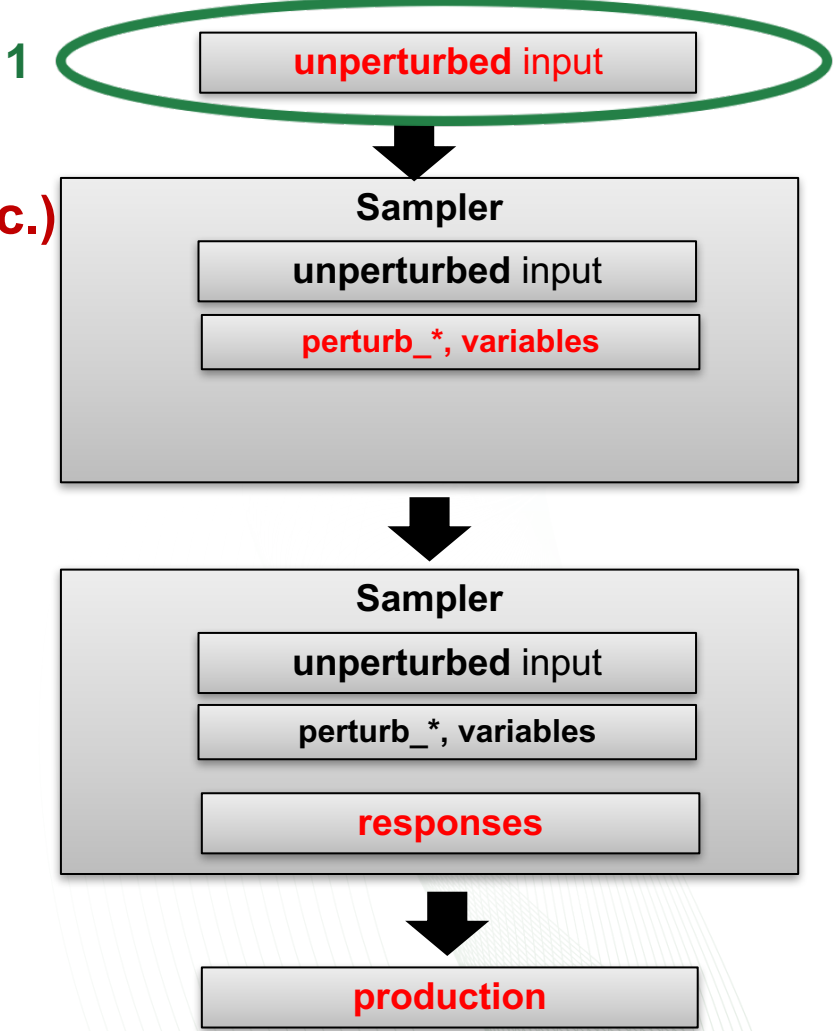
The Sampler Process: Step 1

Create the "unperturbed" SCALE input

- be certain that it **works!**
- **make sure it produces the outputs you want (f71, plt, etc.)**

.out file

```
=====
= Nuclide concentrations in curies for case '1' (#1/1) =
= U238 decay chain =
-----
(relative cutoff; integral of concentrations over time > 1.00E-06 % of integral of all concentrations over time)
.
      0.0E+00y   1.0E+01y   1.0E+03y   1.0E+05y   1.0E+07y   1.0E+09y
tl-206  0.0000E+00  1.4583E-23  1.8040E-16  9.4921E-12  1.0702E-10  9.1781E-11
tl-210  0.0000E+00  3.1186E-20  2.8293E-14  1.4887E-09  1.6784E-08  1.4394E-08
pb-210  0.0000E+00  1.0891E-17  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
pb-214  0.0000E+00  1.4848E-16  1.3470E-10  7.0875E-06  7.9908E-05  6.8531E-05
bi-210  0.0000E+00  1.0891E-17  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
bi-214  0.0000E+00  1.4850E-16  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
po-210  0.0000E+00  1.0891E-17  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
po-214  0.0000E+00  1.4847E-16  1.3470E-10  7.0875E-06  7.9907E-05  6.8530E-05
po-218  0.0000E+00  1.4851E-16  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
at-218  0.0000E+00  2.9701E-20  2.6945E-14  1.4178E-09  1.5985E-08  1.3709E-08
rn-218  0.0000E+00  2.9701E-23  2.6945E-17  1.4178E-12  1.5985E-11  1.3709E-11
rn-222  0.0000E+00  1.4851E-16  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
ra-226  0.0000E+00  1.4851E-16  1.3473E-10  7.0890E-06  7.9924E-05  6.8544E-05
th-230  0.0000E+00  1.0294E-13  1.0351E-09  7.0889E-06  7.9924E-05  6.8544E-05
th-234  0.0000E+00  8.0048E-05  8.0048E-05  8.0047E-05  7.9924E-05  6.8544E-05
pa-234m 0.0000E+00  8.0048E-05  8.0048E-05  8.0047E-05  7.9924E-05  6.8544E-05
pa-234  0.0000E+00  1.2808E-07  1.2808E-07  1.2808E-07  1.2788E-07  1.0967E-07
u-234   0.0000E+00  2.2388E-09  2.2569E-07  1.9692E-05  7.9924E-05  6.8544E-05
u-238   8.0048E-05  8.0048E-05  8.0048E-05  8.0047E-05  7.9924E-05  6.8544E-05
-----
totals  8.0048E-05  2.4028E-04  2.4050E-04  3.3085E-04  1.1191E-03  9.5973E-04
=====
```

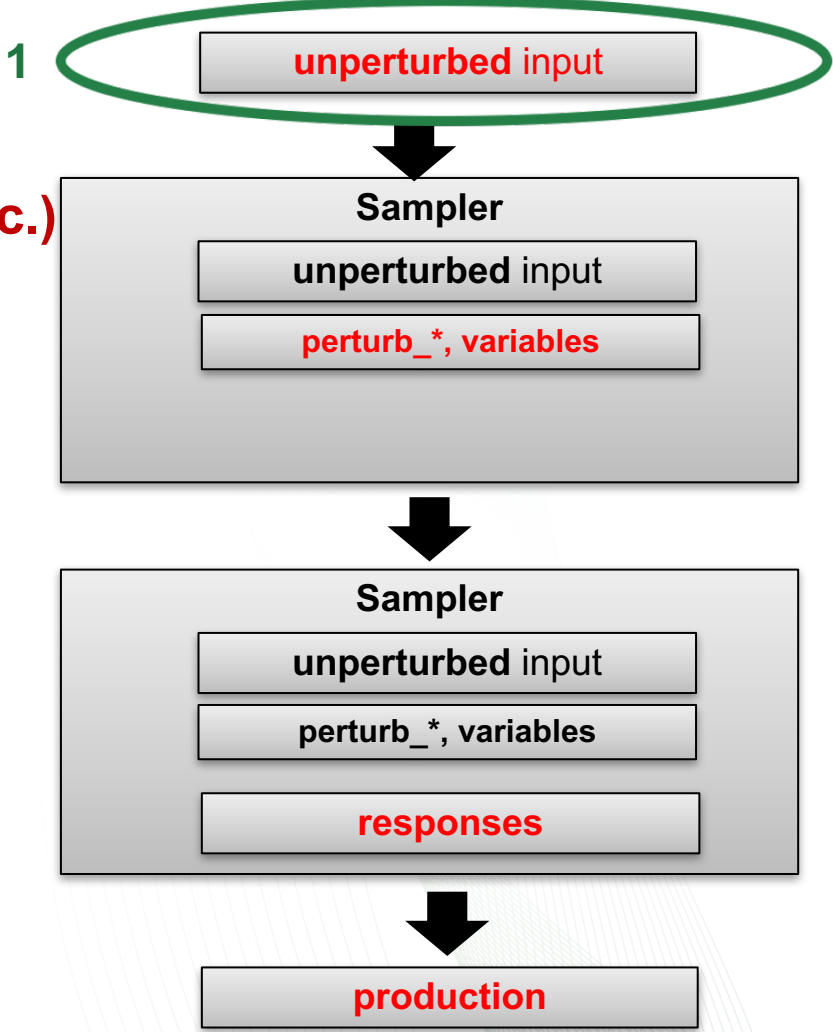
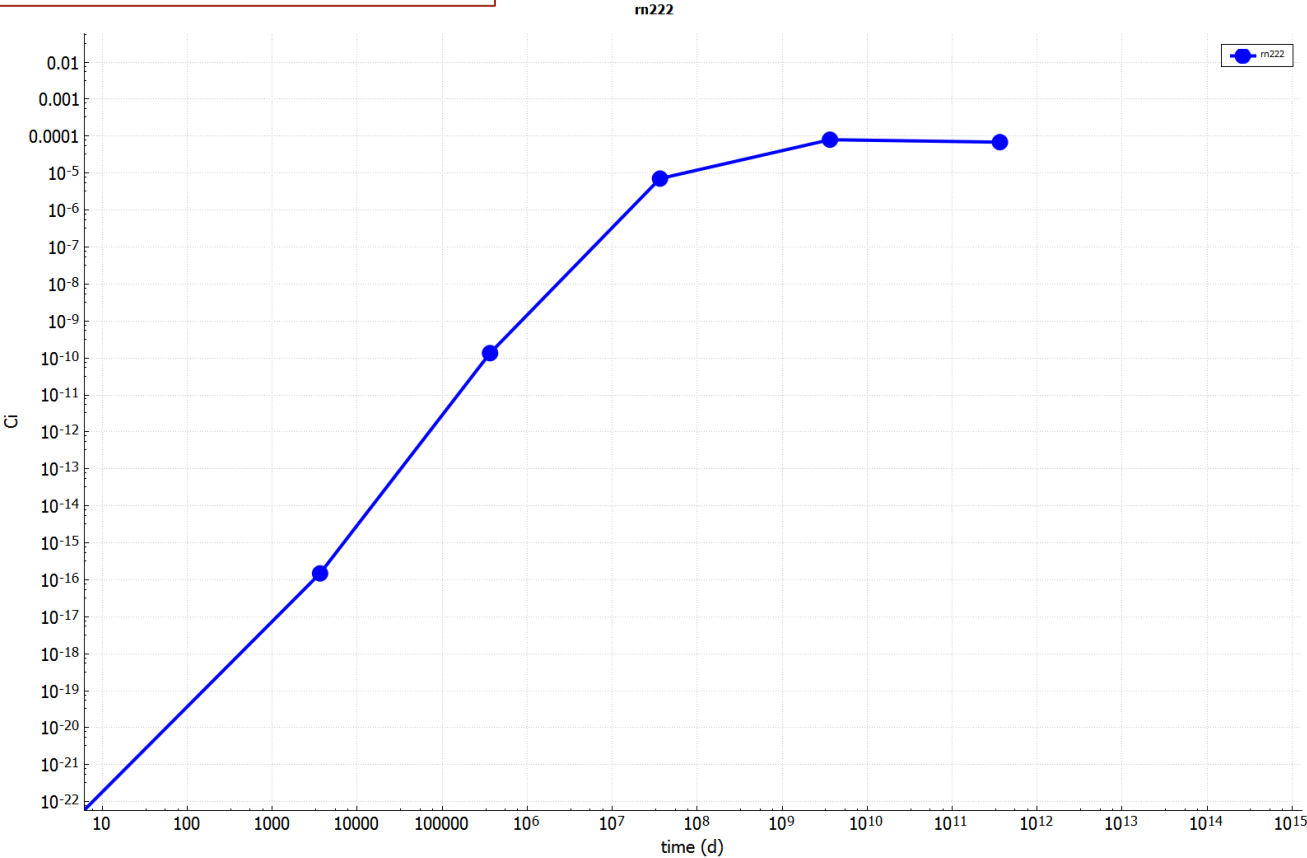


The Sampler Process: Step 1

Create the "unperturbed" SCALE input

- be certain that it works!
- **make sure it produces the outputs you want (f71, plt, etc.)**

.plt file viewed in Fulcrum

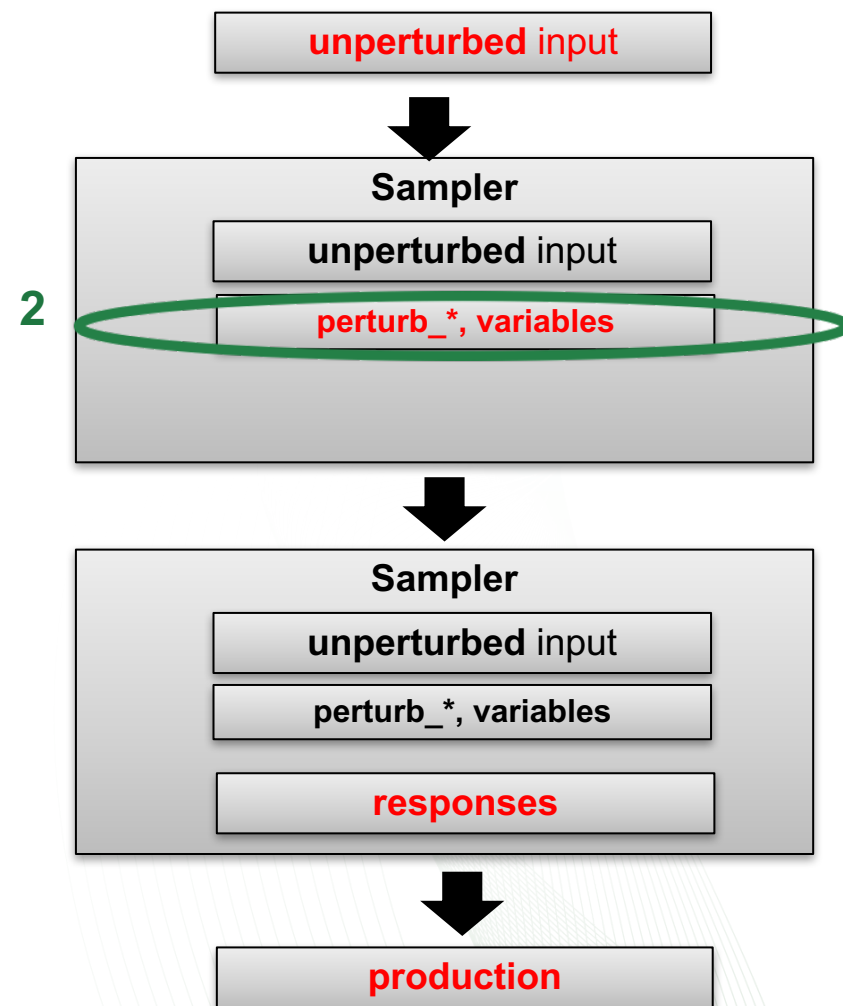


The Sampler Process: Step 2

Wrap it in Sampler

- use `n_samples=2`, `force_run=yes`, `run_cases=no`
- verify you have the desired `perturb_*=yes`
- verify variables are sampled correctly

```
'rn222_uq.starter.inp is a basic sampler template for  
'an ORIGEN followed by an OPUS  
=sampler  
  
read parameters  
'n_samples=100  
'force_run=no  
'run_cases=yes  
'perturb_decay=yes  
'perturb_yields=yes  
'perturb_xs=yes  
'perturb_geometry=yes  
end parameters  
  
read case[c1]  
sequence{PASTE_ORIGEN}sequence  
sequence{PASTE_OPUS}sequence  
end case  
  
end
```



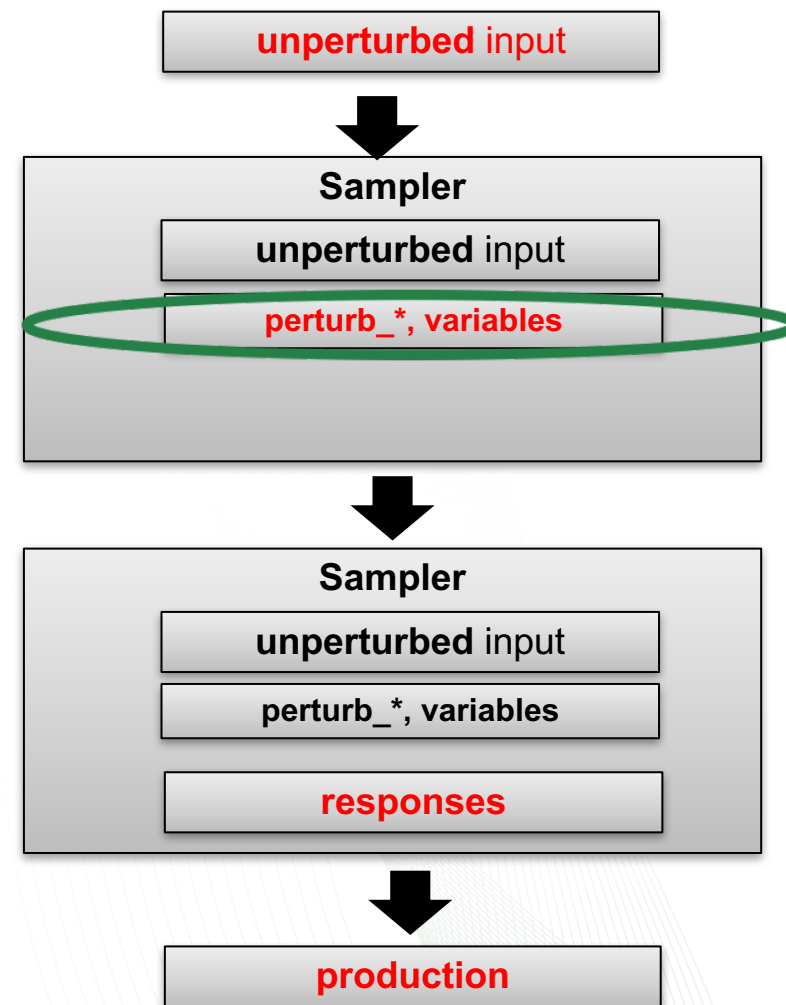
The Sampler Process: Step 2

Wrap it in Sampler

- use `n_samples=2, force_run=yes, run_cases=no`
- verify you have the desired `perturb_*=yes`
- verify variables are sampled correctly

```
'rn222_uq.1.inp shows how to paste sequences in to a single case
=sampler
...
read case[c1]
sequence=origen
case{
  title="U238 decay chain"
  lib{ file="end7dec" }
  time{ units=years t=[3L 1e1 1e9] }
  mat{ iso=[u238=1.0] }
  print{ nuc{ units=[moles curies] } }
  save{ file="ft71f001" steps=[0 1 2 3 4 5] }
}
end sequence
sequence=opus
units=curies
end sequence
end case
end
```

2



The Sampler Process: Step 2

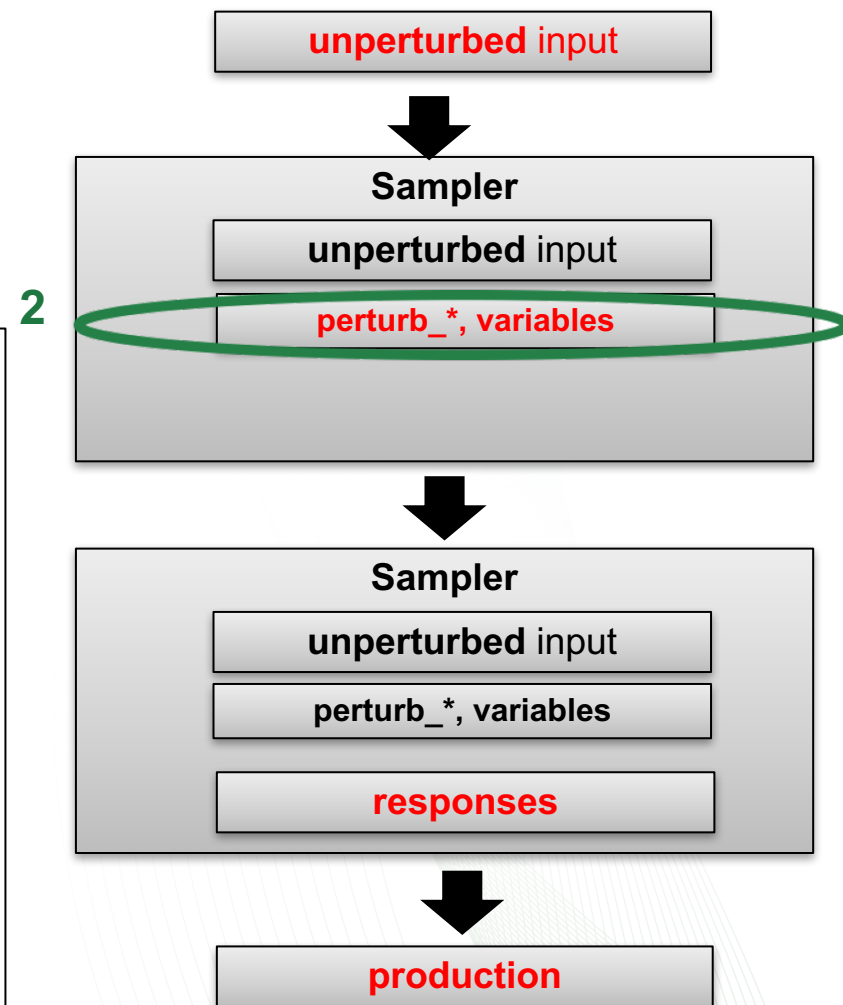
Wrap it in Sampler

- use **n_samples=2, force_run=yes, run_cases=no**
- **verify you have the desired perturb_*=yes**
- verify variables are sampled correctly

```
'rn222_uq.2.inp shows options for variable testing
=sampler
read parameters
n_samples=2
force_run=yes
run_cases=no
perturb_decay=yes
'perturb_yields=yes
'perturb_xs=yes
'perturb_geometry=yes
end parameters

read case[c1]
...
end case

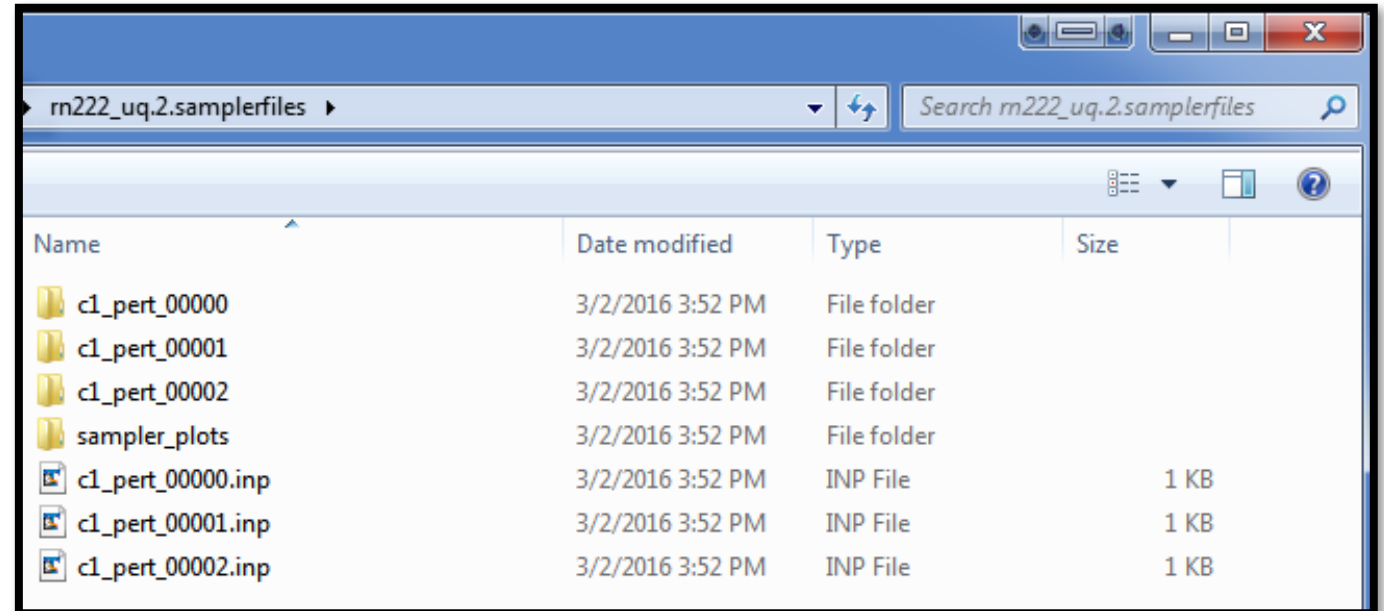
end
```



Sample inputs

sample input naming scheme: **<case_id>_pert_JJJJJ.inp**

- c1 was <case_id>
- 5-digit perturbation index JJJJJ
 - 00000 for unperturbed
 - 00001 for first sample
 - 00002 for second
 - etc.



Sample inputs created by Sampler

compare `c1_pert_00000.inp` and `c1_pert_00001.inp`

```
=origen
case{
  title="U238 decay chain"
  lib{ file="end7dec" }
  time{ units=years t=[3L 1e1 1e9] }
  mat{ iso=[u238=1.0] }
  print{ nuc{ units=[moles curies] } }
  save{ file="ft71f001" steps=[0 1 2 3 4 5] }
}
end
=opus
units=curies
end
```

```
=shell
  ln -sf C:\SCALE-6.2\data\perturb\end7dec_0000 end7dec
end
=origen
case{
  title="U238 decay chain"
  lib{ file="end7dec" }
  time{ units=years t=[3L 1e1 1e9] }
  mat{ iso=[u238=1.0] }
  print{ nuc{ units=[moles curies] } }
  save{ file="ft71f001" steps=[0 1 2 3 4 5] }
}
end
=opus
units=curies
end
```

links in perturbed
decay data

The Sampler Process: Step 3

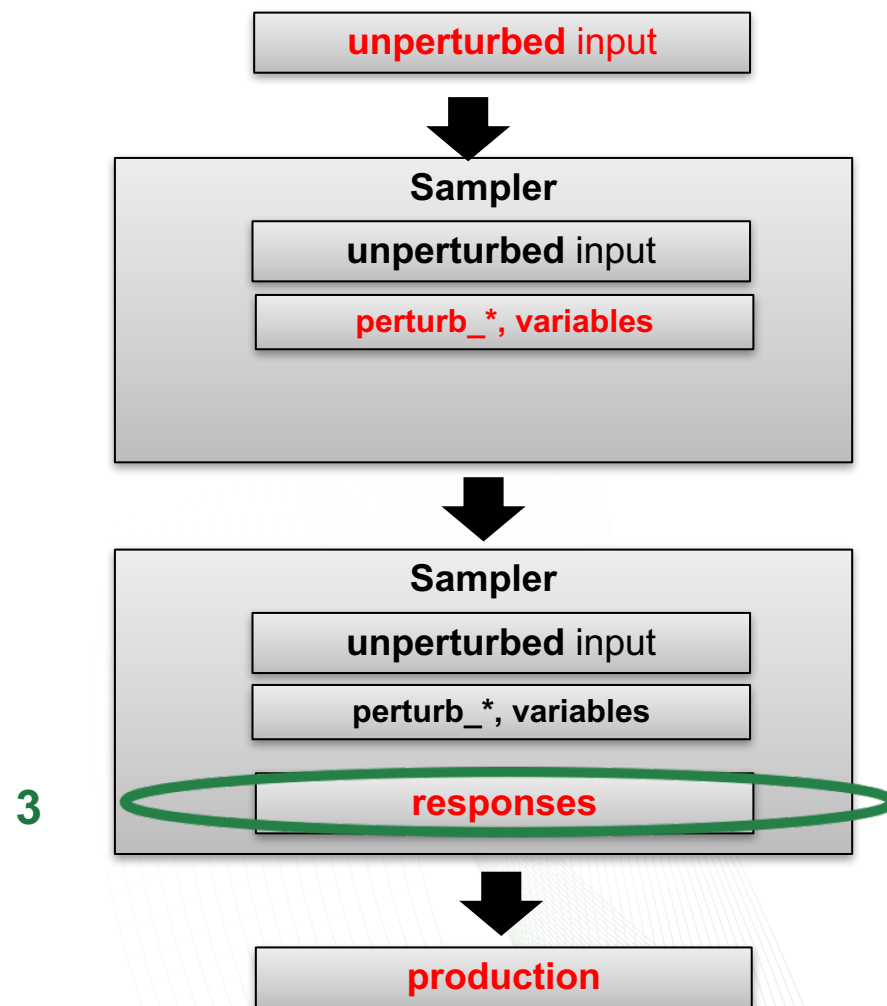
Verify responses

- use `n_samples=2, force_run=no, run_cases=yes`
- NOTE: greps almost always take a few iterations

```
'rn222_uq.3.inp shows options for response testing
=sampler
read parameters
n_samples=2
force_run=no
run_cases=yes
perturb_decay=yes
'perturb_yields=yes
'perturb_xs=yes
'perturb_geometry=yes
end parameters

read case[c1]
...
end case

end
```



Sample outputs created by Sampler

now, when we run with **run_cases=yes**,
we get all the output files
we got for original case,
but now for all samples

**but we need to define
"responses" for
Sampler to process the
output**

Name	Date modified	Type	Size
applet_resources	3/2/2016 4:14 PM	File folder	
c1_pert_00000	3/2/2016 4:14 PM	File folder	
c1_pert_00001	3/2/2016 4:14 PM	File folder	
c1_pert_00002	3/2/2016 4:14 PM	File folder	
sampler_plots	3/2/2016 4:14 PM	File folder	
c1_pert_00000.00000000000000000000.plt	3/2/2016 4:14 PM	PlotOPUS plot file	2 KB
c1_pert_00000.f71	3/2/2016 4:14 PM	F71 File	159 KB
c1_pert_00000.inp	3/2/2016 4:14 PM	INP File	1 KB
c1_pert_00000.msg	3/2/2016 4:14 PM	Outlook Item	1 KB
c1_pert_00000.out	3/2/2016 4:14 PM	OUT File	31 KB
c1_pert_00000.terminal	3/2/2016 4:14 PM	TERMINAL File	3 KB
c1_pert_00001.00000000000000000000.plt	3/2/2016 4:14 PM	PlotOPUS plot file	2 KB
c1_pert_00001.f71	3/2/2016 4:14 PM	F71 File	159 KB
c1_pert_00001.inp	3/2/2016 4:14 PM	INP File	1 KB
c1_pert_00001.msg	3/2/2016 4:14 PM	Outlook Item	1 KB
c1_pert_00001.out	3/2/2016 4:14 PM	OUT File	31 KB
c1_pert_00001.terminal	3/2/2016 4:14 PM	TERMINAL File	3 KB
c1_pert_00002.00000000000000000000.plt	3/2/2016 4:14 PM	PlotOPUS plot file	2 KB
c1_pert_00002.f71	3/2/2016 4:14 PM	F71 File	159 KB
c1_pert_00002.inp	3/2/2016 4:14 PM	INP File	1 KB
c1_pert_00002.msg	3/2/2016 4:14 PM	Outlook Item	1 KB
c1_pert_00002.out	3/2/2016 4:14 PM	OUT File	31 KB
c1_pert_00002.terminal	3/2/2016 4:14 PM	TERMINAL File	3 KB

The Sampler Process: Step 3

Verify responses

- use `n_samples=2, force_run=no, run_cases=yes`
- NOTE: **greps** almost always take a few iterations

```
'rn222_uq.3.inp shows options for response testing
```

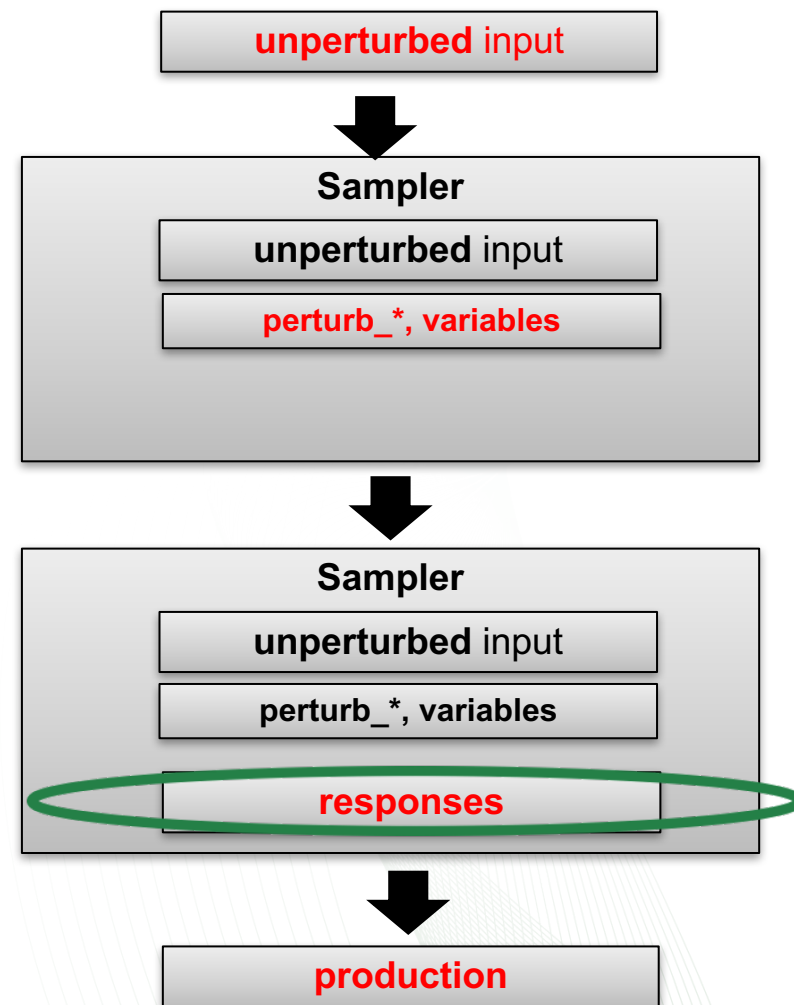
```
=sampler
```

```
...
```

```
'process ft71f001 file  
read response[moles]  
  type=origen_nuclides  
  mixture=1  
  nuclides=rn-222 end  
end response
```

just a
name

```
'process *0.plt file  
read response[curies]  
  type=opus_plt  
  ndataset=0  
  nuclides=rn-222 total end  
end response  
end
```



The Sampler Process: Step 3

Verify responses

- use `n_samples=2, force_run=no, run_cases=yes`
- NOTE: **greps** almost always take a few iterations

rn222_uq.3.out file

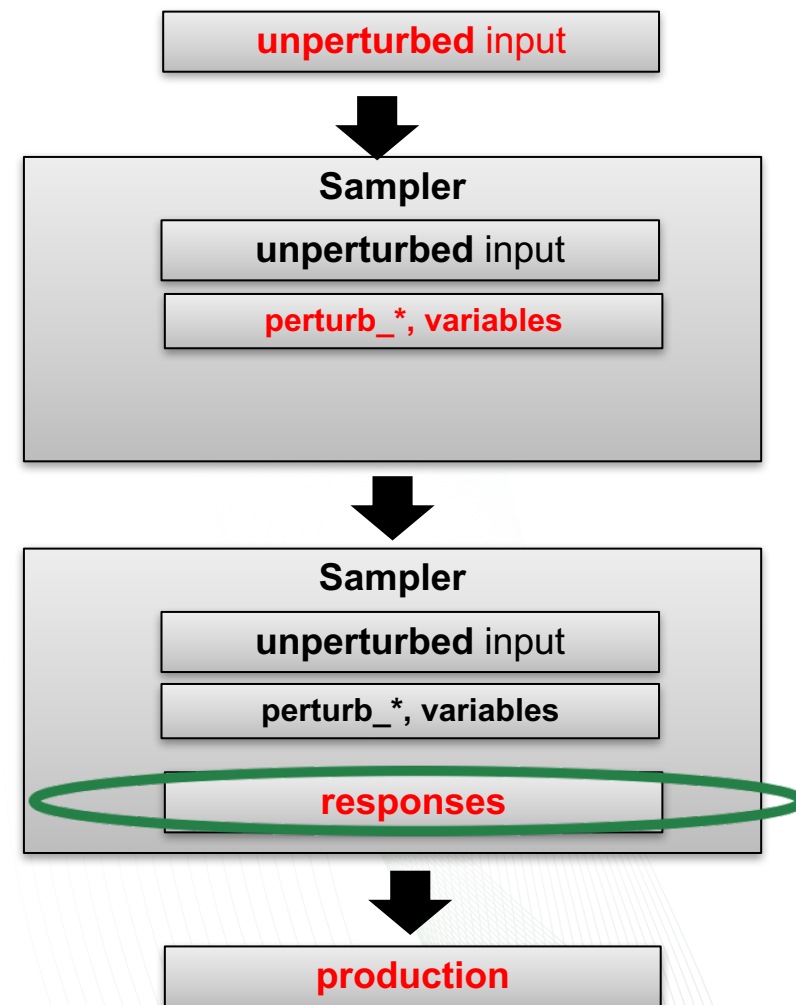
Average + std. deviation table 2 (c1:curies.rn222, c1:curies.total)

time(d)	c1:curies.rn222	c1:curies.total
0	0.0000e+00 +/- 0.0000e+00 (0.00 %)	8.0010e-05 +/- 0.0000e+00 (0.00 %)
3653	1.4780e-16 +/- 4.5461e-19 (0.31 %)	2.4077e-04 +/- 5.4365e-07 (0.23 %)
365300	1.3413e-10 +/- 4.1899e-13 (0.31 %)	2.4100e-04 +/- 5.3541e-07 (0.22 %)
3.653e+07	7.0763e-06 +/- 6.9442e-09 (0.10 %)	3.3117e-04 +/- 5.9067e-07 (0.18 %)
3.653e+09	7.9947e-05 +/- 4.9216e-08 (0.06 %)	1.1197e-03 +/- 2.4944e-06 (0.22 %)
3.653e+11	6.8557e-05 +/- 3.2998e-08 (0.05 %)	9.5987e-04 +/- 2.1203e-06 (0.22 %)

3

IMPORTANT: Do not attempt to interpret uncertainties, yet!
There were only two samples!

The important thing is that we get some result and the process appears to be working as expected.



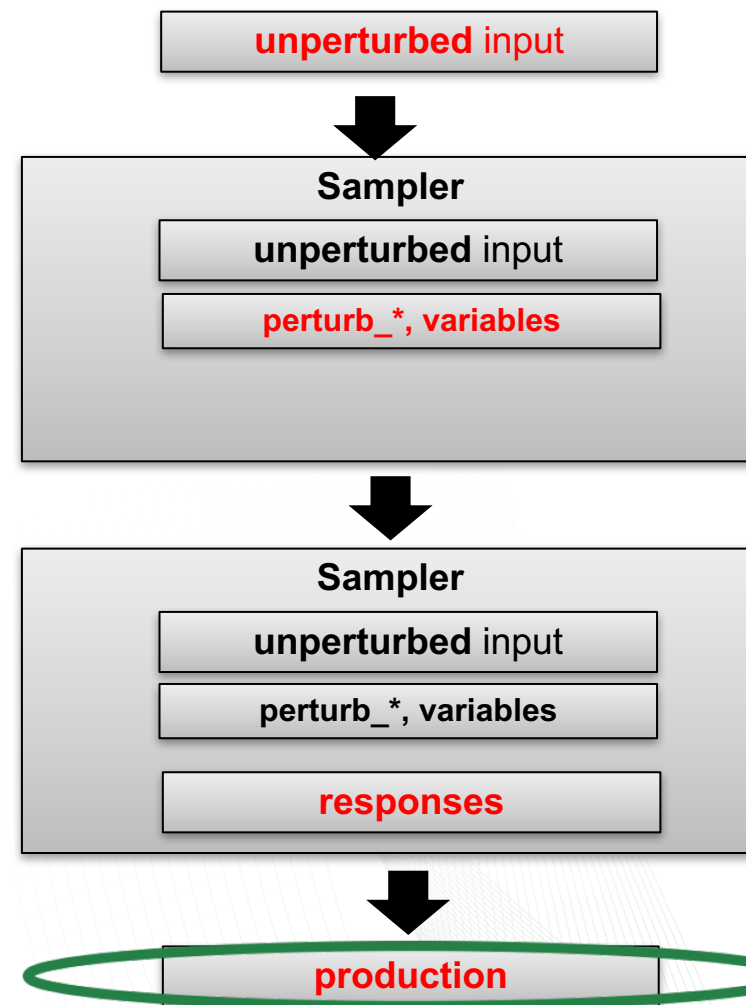
The Sampler Process: Step 4

Start "production" run !!!

- n_samples=100 (or more)!

```
'rn222_uq.final.inp has n_samples=100
=sampler
read parameters
n_samples=100
force_run=no
run_cases=yes
perturb_decay=yes
'perturb_yields=yes
'perturb_xs=yes
'perturb_geometry=yes
end parameters
...
end
```

should take ~3 minutes with 4 threads

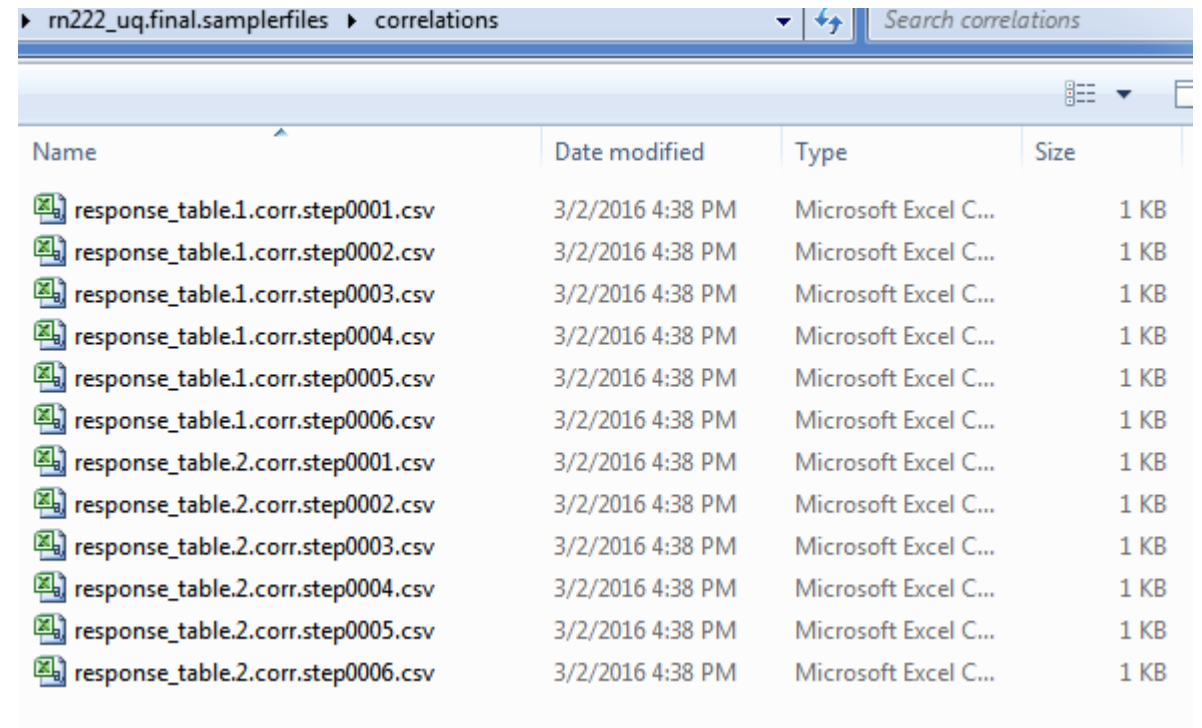


Now analysis begins...

a survey of directory contents

- **correlations/covariances** – csv* sample correlation/sample covariance matrix
 - between each pair of defined responses in a response set
 - do not believe $p \approx 0$ unless you have many samples ($n_{\text{samples}} \approx 1000$)
 - cannot compute if responses do not vary (e.g. for isotopics at time=0)

* our csv is **semi-colon (;) separated** not comma (,) separated to avoid international incidents where "," is used as the decimal e.g. "3,14" instead of "3.14"

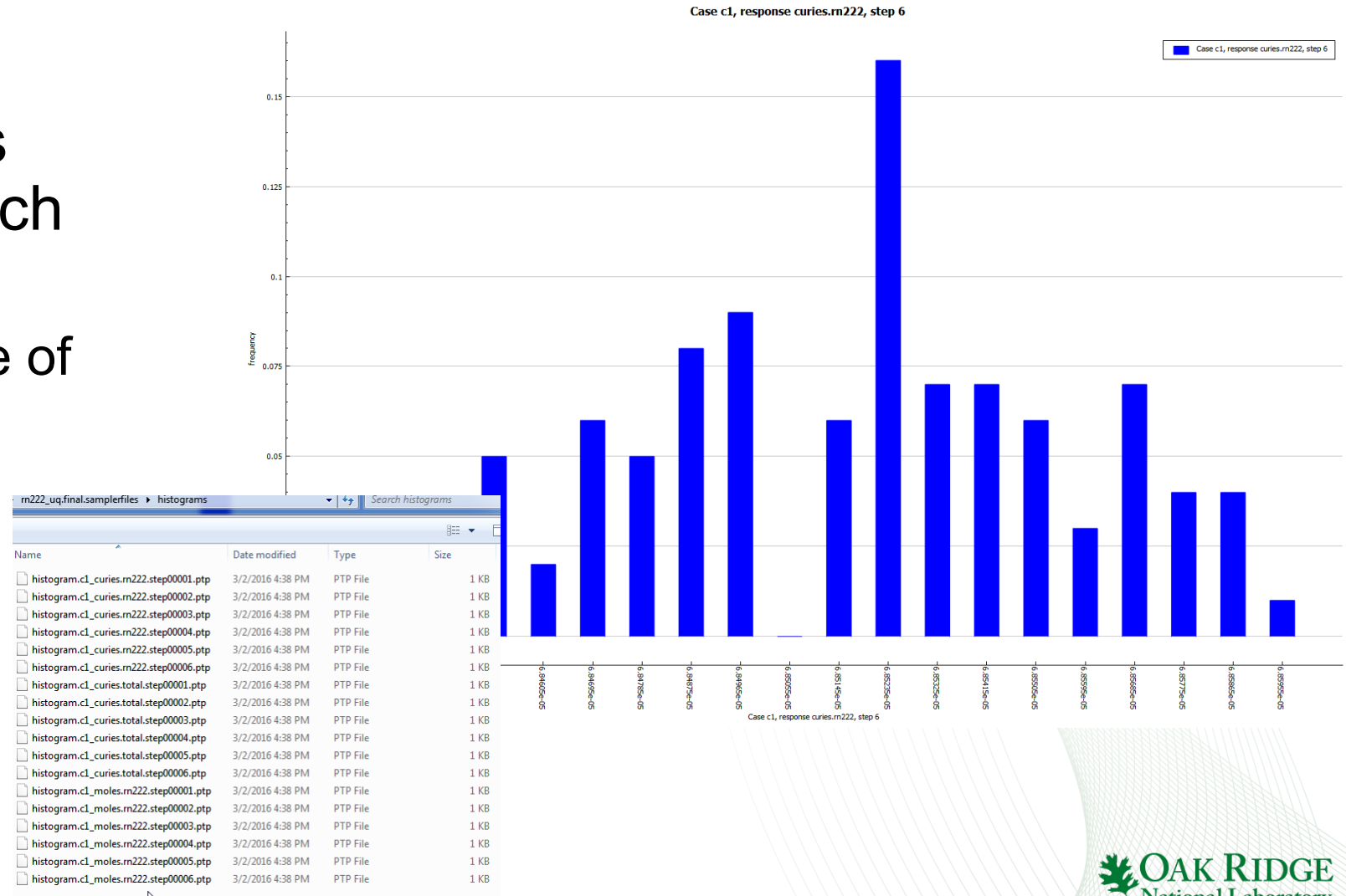


Name	Date modified	Type	Size
response_table.1.corr.step0001.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.1.corr.step0002.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.1.corr.step0003.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.1.corr.step0004.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.1.corr.step0005.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.1.corr.step0006.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0001.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0002.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0003.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0004.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0005.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB
response_table.2.corr.step0006.csv	3/2/2016 4:38 PM	Microsoft Excel C...	1 KB

Now analysis begins...

a survey of directory contents

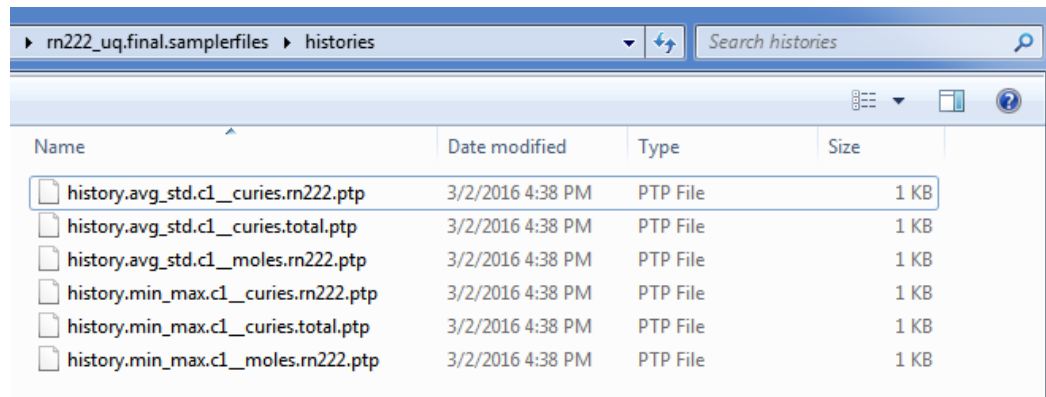
- **histograms** – ptp files with histograms for each response
 - one for each time value of the response
 - 20 bins
 - plottable in Fulcrum (also right-click on plot → save as → image)



Now analysis begins...

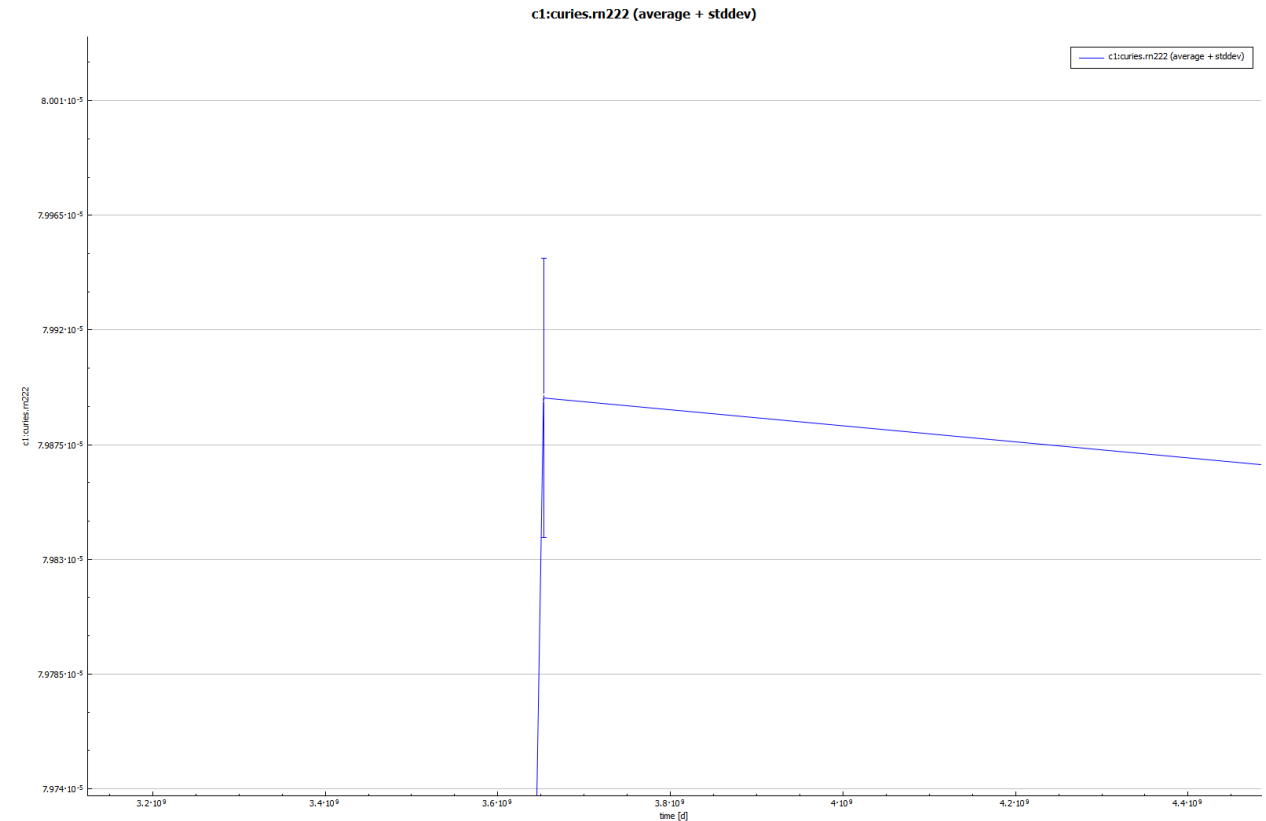
a survey of directory contents

- **histories** – ptp files with time-dependent response
 - average with standard deviation
 - also average with min/max



The screenshot shows a Windows File Explorer window with the address bar set to `rn222_uq.final.samplerfiles > histories`. The search bar contains the text `Search histories`. The main area displays a list of files with the following columns: Name, Date modified, Type, and Size.

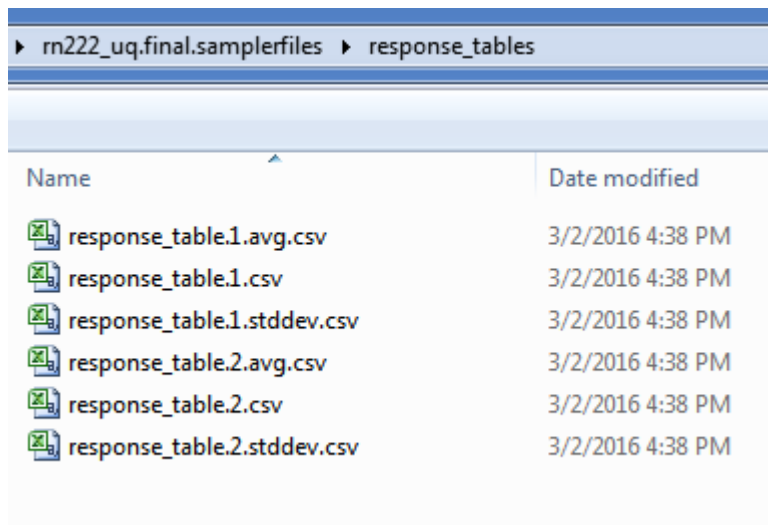
Name	Date modified	Type	Size
history.avg_std.c1_curies.rn222.ptp	3/2/2016 4:38 PM	PTP File	1 KB
history.avg_std.c1_curies.total.ptp	3/2/2016 4:38 PM	PTP File	1 KB
history.avg_std.c1_moles.rn222.ptp	3/2/2016 4:38 PM	PTP File	1 KB
history.min_max.c1_curies.rn222.ptp	3/2/2016 4:38 PM	PTP File	1 KB
history.min_max.c1_curies.total.ptp	3/2/2016 4:38 PM	PTP File	1 KB
history.min_max.c1_moles.rn222.ptp	3/2/2016 4:38 PM	PTP File	1 KB



Now analysis begins...

a survey of directory contents

- **response_tables** – csv files
 - every response
 - every time point
 - versions for sample/standard deviation/average



A	B	C	D	E
sample	time(d)	c1:moles.rn222		
1	0	0.00E+00 +/-	0.00E+00	
1	3652.78	4.35E-24 +/-	0.00E+00	
1	365278	3.94E-18 +/-	0.00E+00	
1	3.65E+07	2.07E-13 +/-	0.00E+00	
1	3.65E+09	2.34E-12 +/-	0.00E+00	
1	3.65E+11	2.01E-12 +/-	0.00E+00	
2	0	0.00E+00 +/-	0.00E+00	
2	3652.78	4.31E-24 +/-	0.00E+00	
2	365278	3.92E-18 +/-	0.00E+00	
2	3.65E+07	2.07E-13 +/-	0.00E+00	
2	3.65E+09	2.34E-12 +/-	0.00E+00	
2	3.65E+11	2.01E-12 +/-	0.00E+00	
3	0	0.00E+00 +/-	0.00E+00	

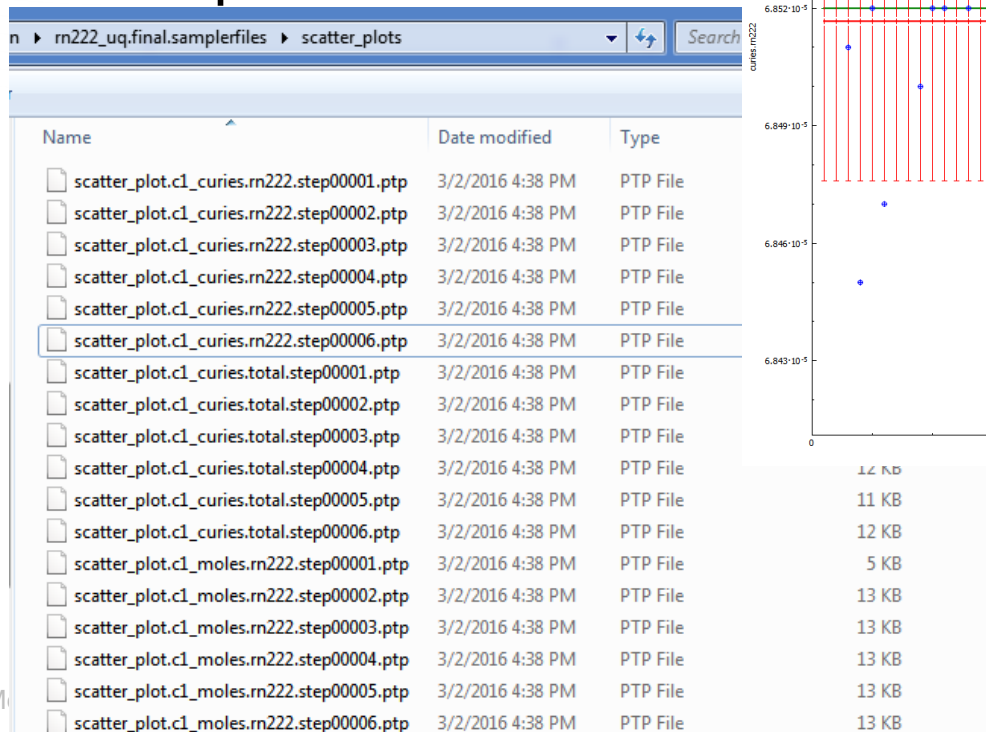
solution
uncertainty

(only populated by
Monte Carlo transport
methods)

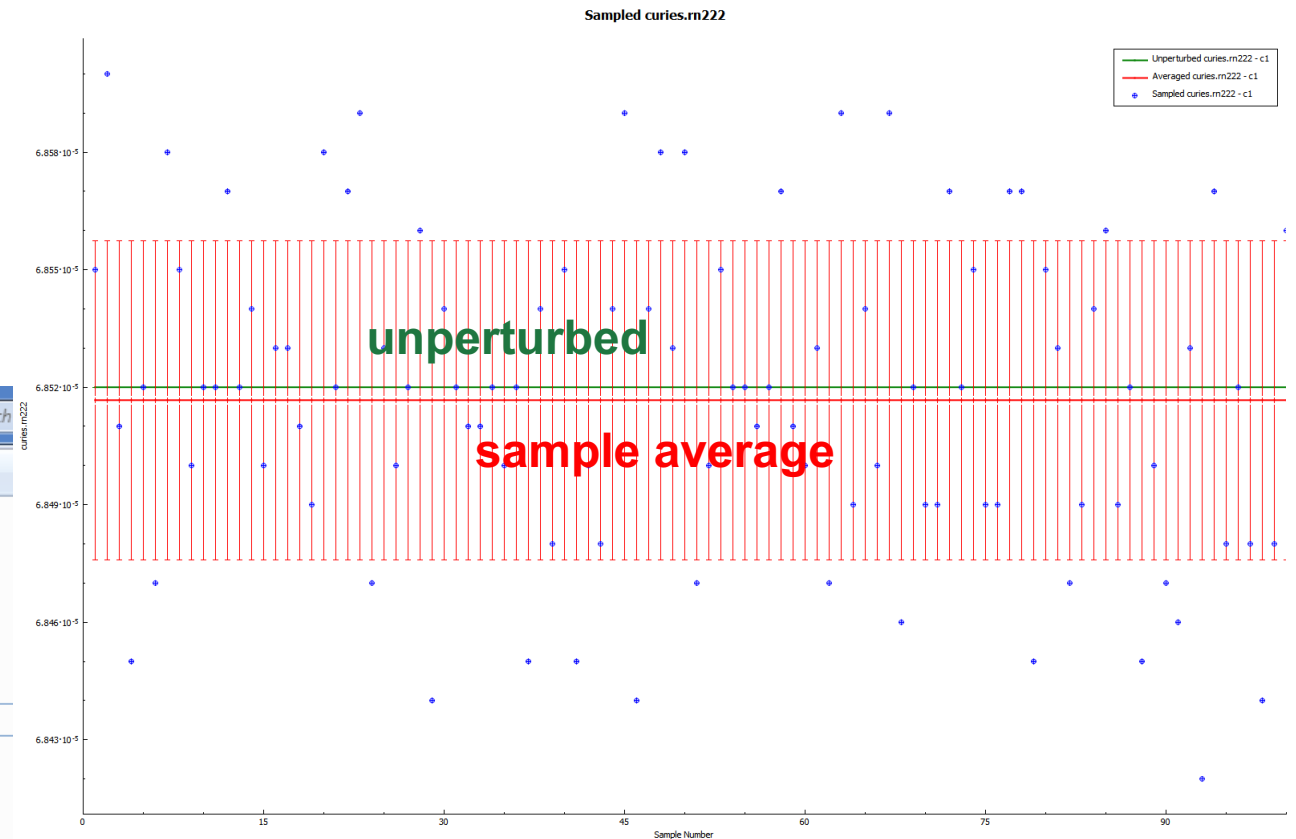
Now analysis begins...

a survey of directory contents

- **scatter_plots** – ptp files showing data points
 - average vs. unperturbed!



Name	Date modified	Type	Size
scatter_plot.c1_curies.rm222.step00001.ptp	3/2/2016 4:38 PM	PTP File	12 KB
scatter_plot.c1_curies.rm222.step00002.ptp	3/2/2016 4:38 PM	PTP File	11 KB
scatter_plot.c1_curies.rm222.step00003.ptp	3/2/2016 4:38 PM	PTP File	12 KB
scatter_plot.c1_curies.rm222.step00004.ptp	3/2/2016 4:38 PM	PTP File	5 KB
scatter_plot.c1_curies.rm222.step00005.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.rm222.step00006.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00001.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00002.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00003.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00004.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00005.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_curies.total.step00006.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00001.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00002.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00003.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00004.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00005.ptp	3/2/2016 4:38 PM	PTP File	13 KB
scatter_plot.c1_moles.rm222.step00006.ptp	3/2/2016 4:38 PM	PTP File	13 KB



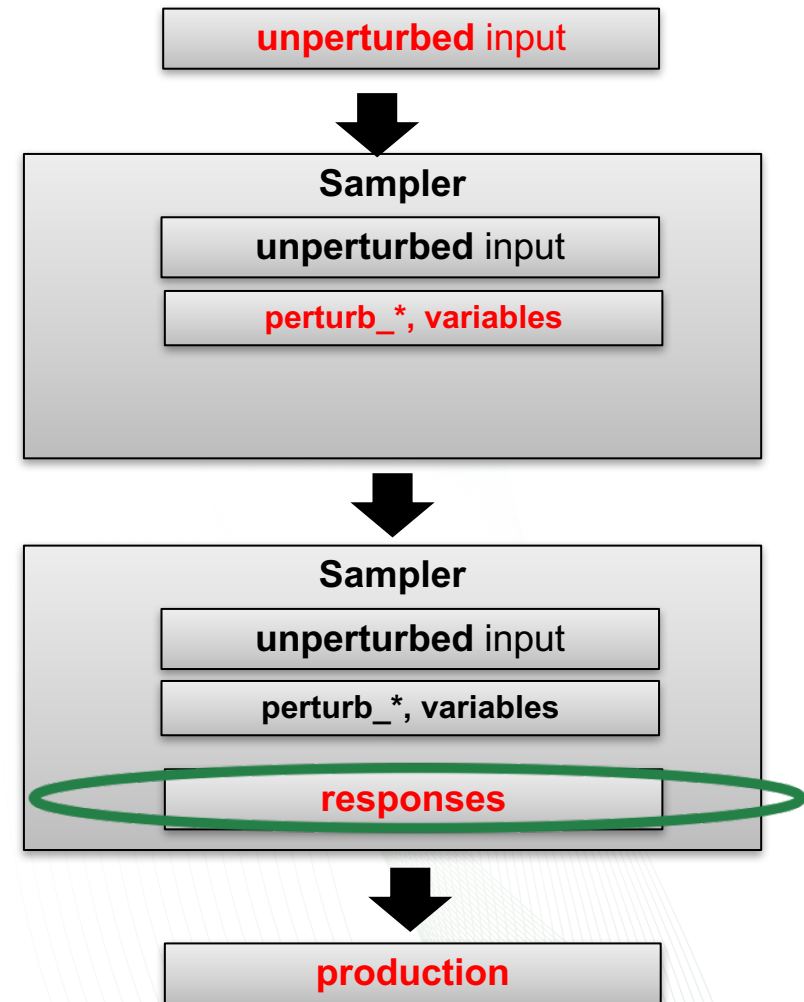
And the answer is...

Uncertainty peaks at about 0.7% but decreases to 0.07% at long times (~1 billion years)

rn222_uq.final.out file

Average + std. deviation table 2 (c1:curies.rn222, c1:curies.total)

time(d)	c1:curies.rn222			c1:curies.total		
0	0.0000e+00	+/- 0.0000e+00	(0.00 %)	8.0010e-05	+/- 0.0000e+00	(0.00 %)
3653	1.4835e-16	+/- 1.0360e-18	(0.70 %)	2.4042e-04	+/- 1.1034e-06	(0.46 %)
365300	1.3460e-10	+/- 9.0120e-13	(0.67 %)	2.4065e-04	+/- 1.1069e-06	(0.46 %)
3.653e+07	7.0849e-06	+/- 2.7948e-08	(0.39 %)	3.3090e-04	+/- 1.1306e-06	(0.34 %)
3.653e+09	7.9893e-05	+/- 5.4575e-08	(0.07 %)	1.1183e-03	+/- 2.9942e-06	(0.27 %)
3.653e+11	6.8517e-05	+/- 4.0519e-08	(0.06 %)	9.5910e-04	+/- 2.5288e-06	(0.26 %)



Some comments on Decay Uncertainty

- Uncertainty is in general small for decay constants/branching ratios
 - exception is short-lived fission products, e.g. following burst fission
- Including decay uncertainty in a calculation requires direct usage of the "end7dec" ORIGEN decay library
 - COUPLE and ORIGEN if they use "end7dec"
 - pre-generated ORIGEN reactor libraries have embedded decay constants
 - **TRITON and Polaris do not use end7dec by default**

```
=shell
  ln -sf C:\SCALE-6.2\data\perturb\end7dec_0000 end7dec
end

=origen
case{
  title="U238 decay chain"
  lib{ file="end7dec" }
  time{ units=years t=[3L 1e1 1e9] }
  mat{ iso=[u238=1.0] }
  print{ nuc{ units=[moles curies] } }
  save{ file="ft71f001" steps=[0 1 2 3 4 5] }
}
end
=opus
units=curies
end
```

Sampler Tutorial

Variables, Placeholders, and Responses



Topics

- How to define variables
 - distributions (normal, uniform, beta)
 - expressions
- How to insert variables into an input using the "placeholder" approach
- How to define responses

Variables: Normal and Uniform

- general form

```
read variable[ID]  
  distribution=TYPE  
  cases= c1 c2 ... end  
  [distribution-specific-params]  
end variable
```

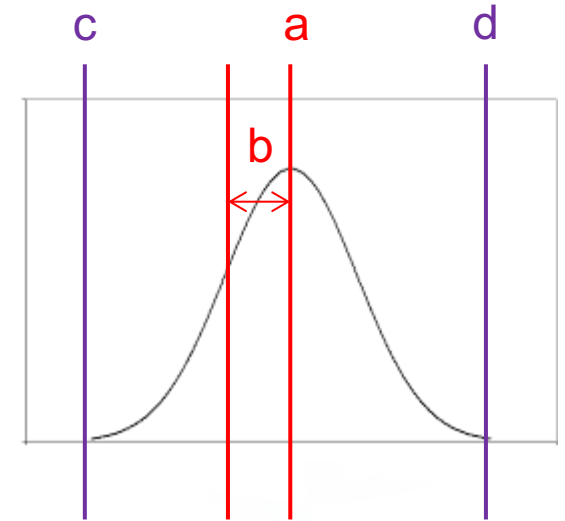
- placeholder usage
(*anywhere in input*)

```
uo2 ... u235 #{ID} u238
```

Normal Distribution

```
read variable[ID]  
  distribution=normal  
  value=a  
  stddev=b  
  minimum=c  
  maximum=d  
  cases=c1 end  
end variable
```

NOTE: value c, d
are used to
truncate

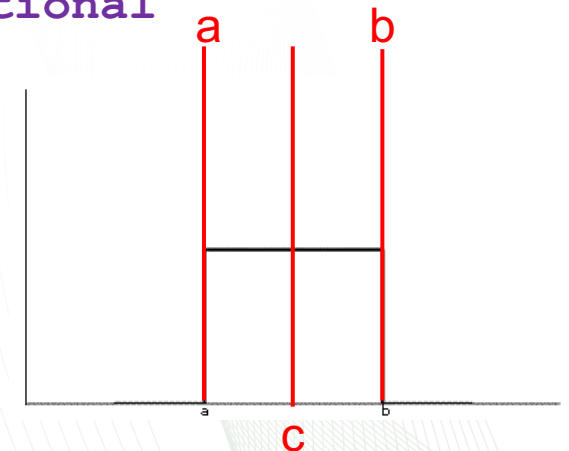


required
optional

Uniform Distribution

```
read variable[ID]  
  distribution=uniform  
  minimum=a  
  maximum=b  
  value=c  
  cases=c1 end  
end variable
```

NOTE: value c
will be
unperturbed value



Variables: Beta

- general form

```
read variable[ID]
  distribution=TYPE
  cases= c1 c2 ... end
  [distribution-specific-params]
end variable
```

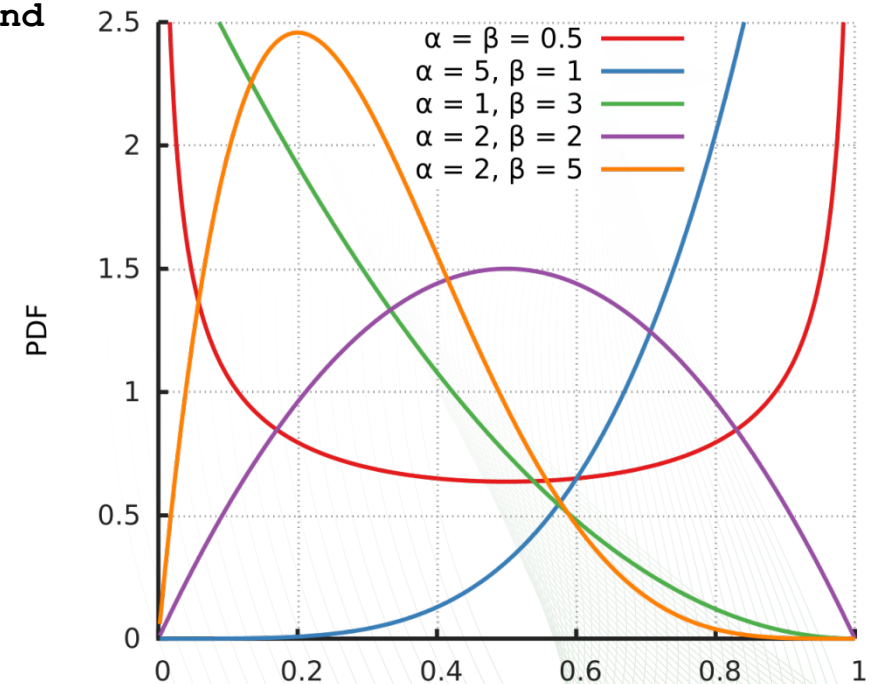
- placeholder usage
(*anywhere in input*)

u02 ... u235 **#{ID}** u238

Beta Distribution

```
read variable[ID]
  distribution=beta
  beta_a= $\alpha$ 
  beta_b= $\beta$ 
  minimum=c
  maximum=d
  value=e
  cases= c1 end
end variable
```

NOTE: α , β , c, d
define distribution
and value e will be
used as unperturbed
value



Variables: Expressions

- general form

```
read variable[ID]  
  distribution=TYPE  
  cases= c1 c2 ... end  
  [distribution-specific-params]  
end variable
```

- placeholder usage
(*anywhere in input*)

u02 ... u235 **#{ID}** u238

Expression

```
read variable[a]  
...  
end variable  
read variable[b]  
...  
end variable
```

```
read variable[ID]  
  distribution=expression  
  expression="53-cos (a) +exp (b) /2"  
  cases= c1 end  
end variable
```

Allowed Expressions

Operator	Description	Example
+	Addition	3+3, x+y
+	Unary Plus	+3, +(-3), +x
-	Subtraction	3-3, x-y
-	Unary Minus	-3, -(3+3), -x
*	Multiplication	3*3, x*y
/	Division	3/3, x/y
%	Modulo	3%3, x%y
^	Power	3^3, x^y
==	Equal to	3==3, x==y
!=	Not equal to	3!=3, x!=y
<	Less than	3<3, x<y
>	Greater than	3>3, x>y
<=	Less than or equal to	3<=3, x<=y
>=	Greater than or equal to	3>=3, x>=y
and	Left is true and right is true	1 and 1, x==y and y==z
or	Left is true or right is true	1 or 1, x==y or y==z
nand	Not and	1 nand 1, 1 nand 0, x nand y
xor	Exclusive or	1 xor 1, 1 xor 0, x xor y
nor	Not or	1 nor 1, 1 nor 0, x nor y
abs(x)	Absolute value	abs(1),abs(-1),abs(x),abs(y)
min(x,...n)	Minimum of values	min(1),min(1,2,3,4,5,x,y),etc
max(x,...n)	Maximum of values	max(1),max(1,2,3,4,5,x,y),etc
avg(x,...n)	Average of values	avg(1),avg(1,2,3,4,5,x,y),etc
Sum(x,...n)	Summation of values	sum(1),sum(1,2,3,4,5,x,y),etc
mul(x,...n)	Multiply values	mul(1),mul(1,2,3,4,5,x,y),etc
floor(x)	Floor of the value	floor(1)=1, floor(-1.0)=-1.0, floor(-1.1)=-2.0

Operator	Description	Example
floor(x)	Floor of the value	floor(1)=1, floor(-1.0)=-1.0, floor(-1.1)=-2.0
ceil(x)	Ceiling of the value	ceil(1)=1,ceil(-1.1)=-1
round(x)	Round the value to nearest integer	round(1.1)=1,round(1.5)=2,round(1.9)=2
roundn(x,n)	Round the value to the nth decimal position	roundn(1.15,1)= 1.2, roundn(1.519,2)= 1.52
cos(x)	Cosine of the value	cos(0.0)=1
sin(x)	Sine of the value	sin(0.0)=0
tan(x)	Tangent of the value	tan(0.0)=0
sec(x)	Secant of the value	sec(0.0)=1
csc(x)	Cosecant of the value	csc(1.0)= 1.1884
cot(x)	Cotangent of the value	cot(1.0)= 0.642093
sinh(x)	Hyperbolic sine of the value	sinh(1.0)= 1.1752
cosh(x)	Hyperbolic cosine of the value	cosh(1.0)= 1.54308
tanh(x)	Hyperbolic tangent of the value	tanh(1.0)= 0.761594
log(x)	Natural log of the value	log(2.7182818)=1
log10(x)	Common log of the value	log10(10.0)=1
exp(x)	Exponential of the value	exp(1)=2.7182818
clamp(l,x,u)	Clamp the value x between lowerbound l and upperbound u	clamp(-1,1,+1)=1,clamp(-1,-5,+1)=-1
inrange(l,x,u)	Determine if value x is greater than or equal to lowerbound l, and less than or equal to upperbound u	inrange(-1,1,+1)=1,inrange(-1,-5,+1)=0
sgn(x)	Determine the sign of x.	sgn(+1)=1,sgn(-1)=-1,sgn(0)=0,sgn(5)=1
sqrt(x)	Square Root of the value	sqrt(1)=1,sqrt(4)=2
root(x,y)	Root of x given y	root(4,2)=2,root(8,3)=2
true	Constant true value	true
false	Constant false value	False
if(condition,true,false)	If function that evaluates the condition. If true, the true value is substituted, else the false value is substituted	if(x > y, x, y) = x

Placeholder Example

- Placeholders are variable IDs inside # {}, e.g. `# {NORMAL}`
- must have perturb_geometry=yes
- Using `shell` is an easy way to test the sampling process

```
=sampler
  read parameters
    n_samples = 5
    perturb_bondarenko = no
    perturb_pointwise = no
    perturb_xs = no
    perturb_geometry = yes
    force_run = yes
  end parameters

  read case[c1]
    sequence=shell
    echo NORMAL= # {NORMAL}
    echo UNIFORM= # {UNIFORM}
  end sequence
end case

read variable[NORMAL]
  distribution=normal
  minimum=-2 value=-1 maximum=0 stddev=2
  cases=c1 end
end variable

read variable[UNIFORM]
  distribution=uniform
  minimum=-2 value=-1 maximum=0
  cases=c1 end
end variable

end
```

Responses

- general form

```
read response[NAME]  
  type=TYPE  
  [type-specific-params]  
end variable
```

- **TYPE**

- **opus_plt** – data from an OPUS-generated PLT file
- **triton** - TRITON homogenized cross-sections (xfile016)
- **stdcmp** – standard composition files
- **f71** – concentrations from the F71 ORIGEN dump
- **grep** – general expression from the text output file
- **variables** – the geometry perturbation sampled values

Responses: Grep

- most general in the sense that it just applies the searching utility "grep" to the OUTPUT file for each sample
- Uses regular expressions with POSIX classes, e.g. "[[:digit:]]"
- Special "scale.number:" shortcut matches any number (1e3, 1.23, -3, etc.)
- uses "grep -o" which means only return matching part
- multiple regexp are allowed (usually need 2—one to match the line of interest and one to isolate the number)
- each regexp takes result of previous regexp as input

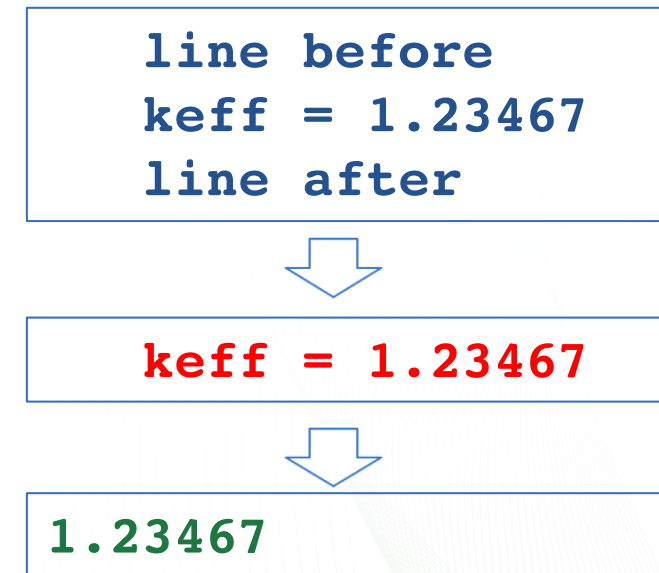
Grep Example

- inside .out file somewhere
- grep to extract

```
read variable[keff]
  distribution=grep
  regexp="keff = [[:digit:]]\.[[:digit:]]*"
  regexp="[[:digit:]]\.[[:digit:]]*"
end variable
```

- alternative using ":scale.number:"

```
read variable[keff]
  distribution=grep
  regexp="keff = :scale.number:"
  regexp=":scale.number:"
end variable
```



Grep Response Example

- add these responses to previous example

```
read response[RNORMAL]
  type=grep
  regexp="NORMAL= *:scale.number:"
  regexp=":scale.number:"
end response

read response[RUNIFORM]
  type=grep
  regexp="UNIFORM= *:scale.number:"
  regexp=":scale.number:"
end response
```


Summary

- variables
 - defines a distribution (*normal, uniform, beta, expression*)
 - **syntax:** read variable[id] ... end variable
 - placeholder with #{id} is easiest way to "insert" into input
- responses
 - defines a quantity of interest (QOI)
 - many types including "grep" for scraping output file

```
read variable[NORMAL]
  distribution=normal
  minimum=-2 value=-1 maximum=0 stddev=2
  cases=Case1 end
end variable

read variable[UNIFORM]
  distribution=uniform
  minimum=-2 value=-1 maximum=0
  cases=Case1 end
end variable

read response[RNORMAL]
  type=grep
  regexp="NORMAL= *:scale.number:"
  regexp=":scale.number:"
end response

read response[RUNIFORM]
  type=grep
  regexp="UNIFORM= *:scale.number:"
  regexp=":scale.number:"
end response

read response[VNORMAL]
  type=variables
  data=NORMAL end
end response
```