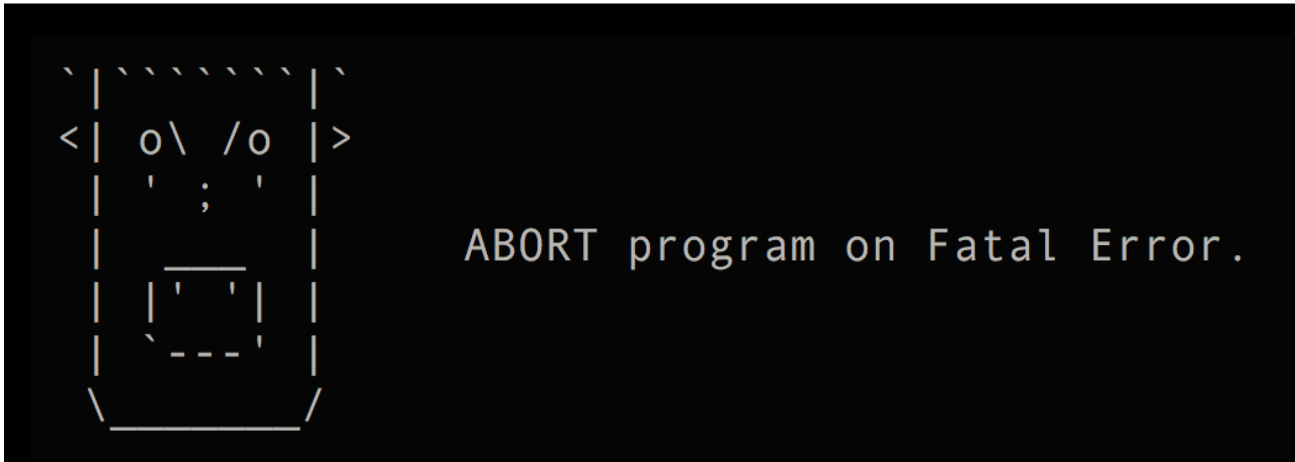


```
`|```````|`
<| o\ /o |>
 | ' ; ' |
 |  ___  |
 | |' '| |
 | `---' |
 \_______/
```

```
ABORT program on Fatal Error.
```

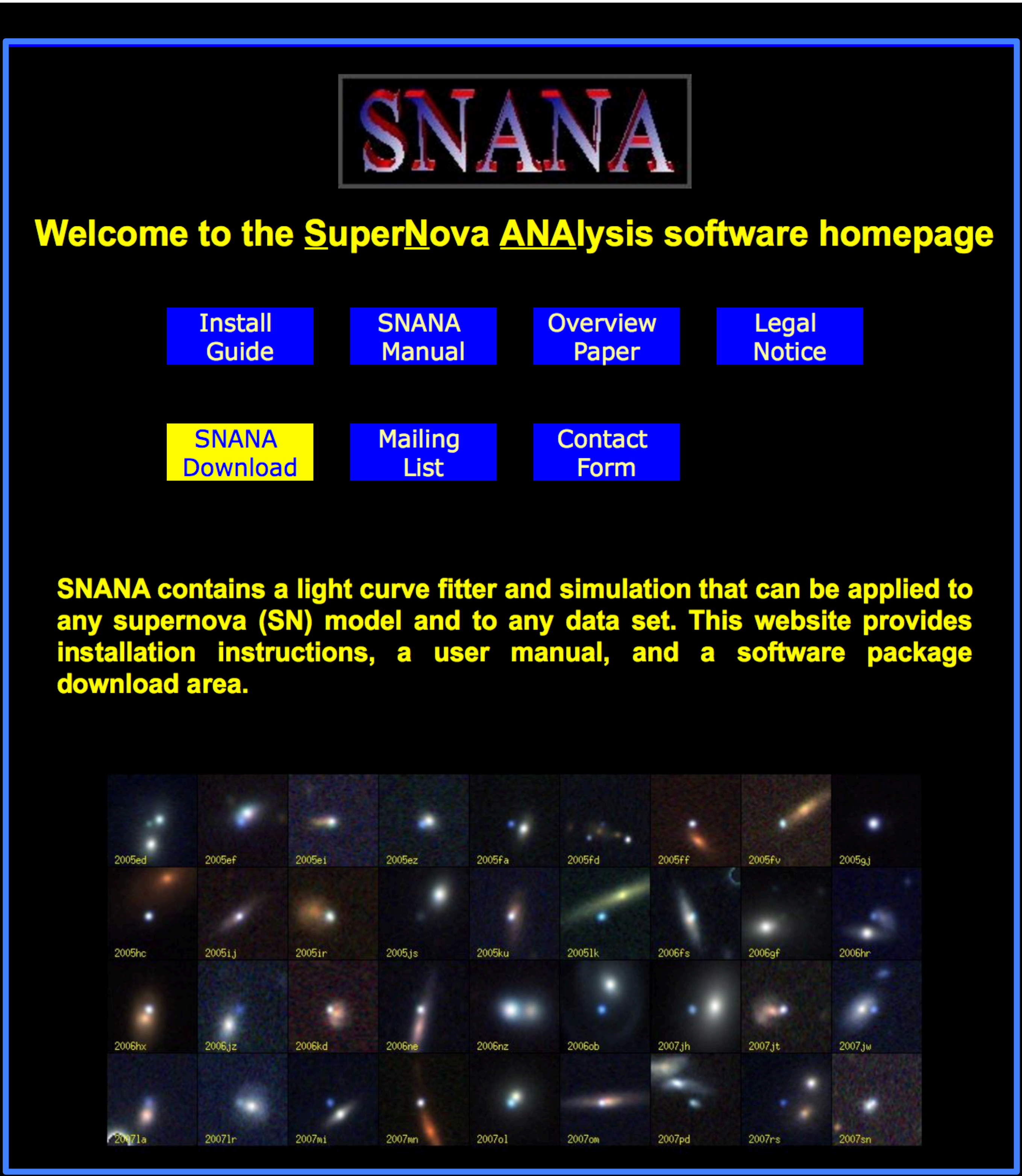
Kavli Institute for Cosmological Physics AT THE UNIVERSITY OF CHICAGO

SNANA Tutorial

R. Kessler

May 2016

GOOGLE Search: No, not this SNANA



SNANA

Welcome to the SuperNova ANAlysis software homepage

SNANA contains a light curve fitter and simulation that can be applied to any supernova (SN) model and to any data set. This website provides installation instructions, a user manual, and a software package download area.

This one

Outline

- Architecture
- Software Interface for Private Code
- Systematics & Multi-Core Jobs
- Output
- Documentation

Architecture: Summary of Ready-to-Run Programs in $SNANA_DIR

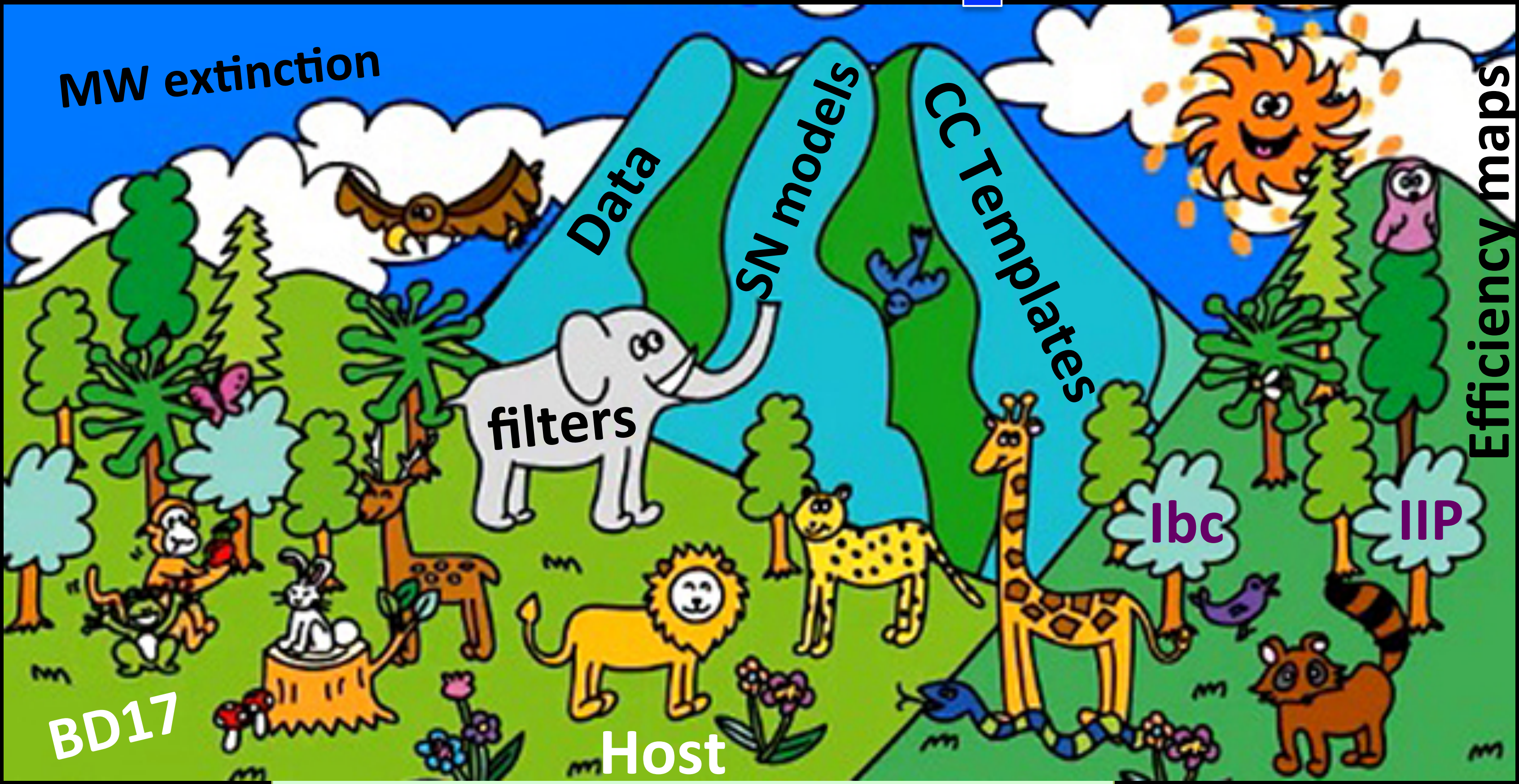
- Simulation package (catalog, not pixels)
- Light Curve Fitting & Template Fitting
- Hubble Diagram Fitting

  (old, simple, fast: better codes elsewhere)
- Utilities for systematics & multi-core processing
- *NO Image-Processing Tools*

Architecture: Environment

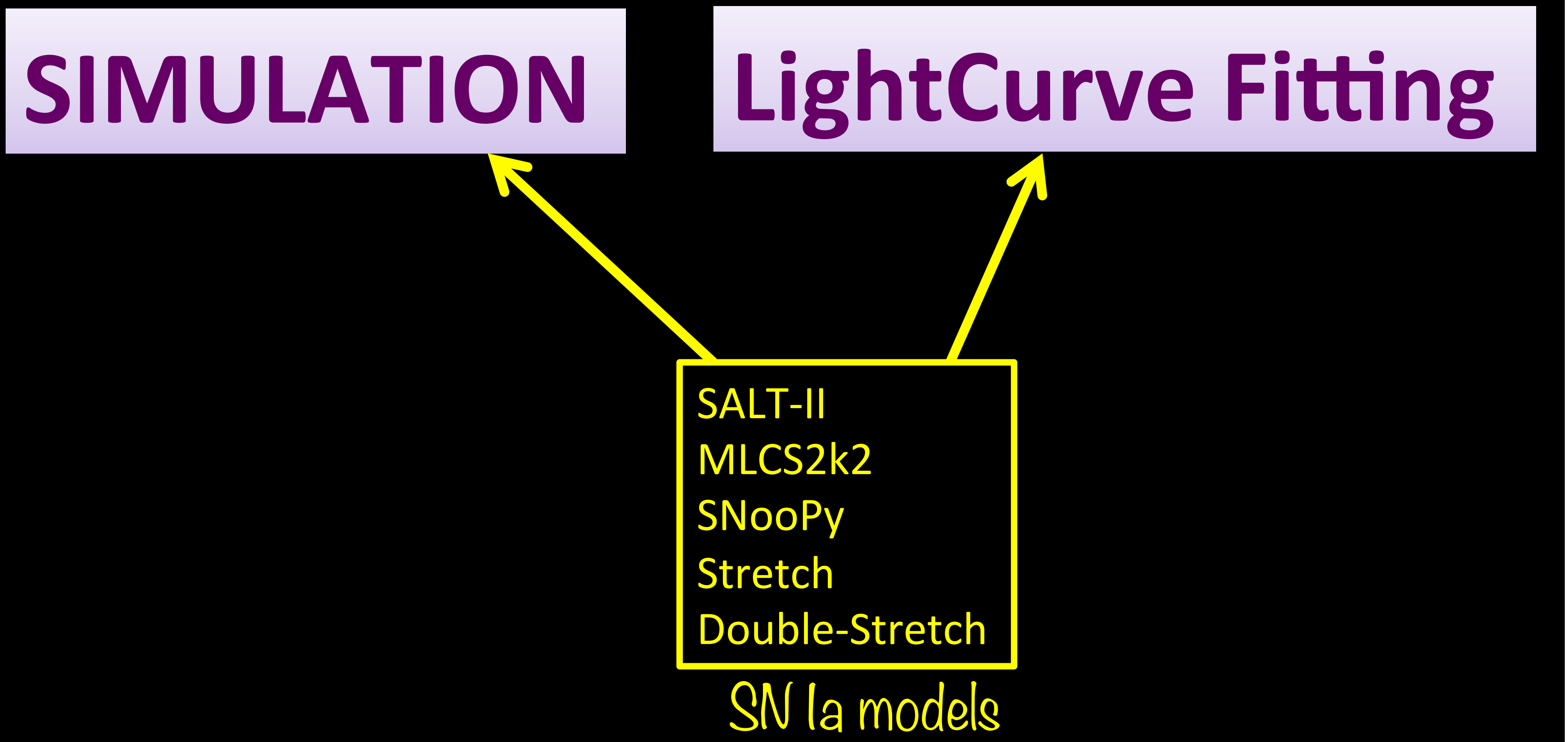
- Simulation package (catalog, not pixels)
- Light Curve Fitting & Template Fitting
- Hubble Diagram Fitting
- Utilities for systematics & multi-core processing

$SNANA_DIR



$SNDATA_ROOT

SNANA Architecture



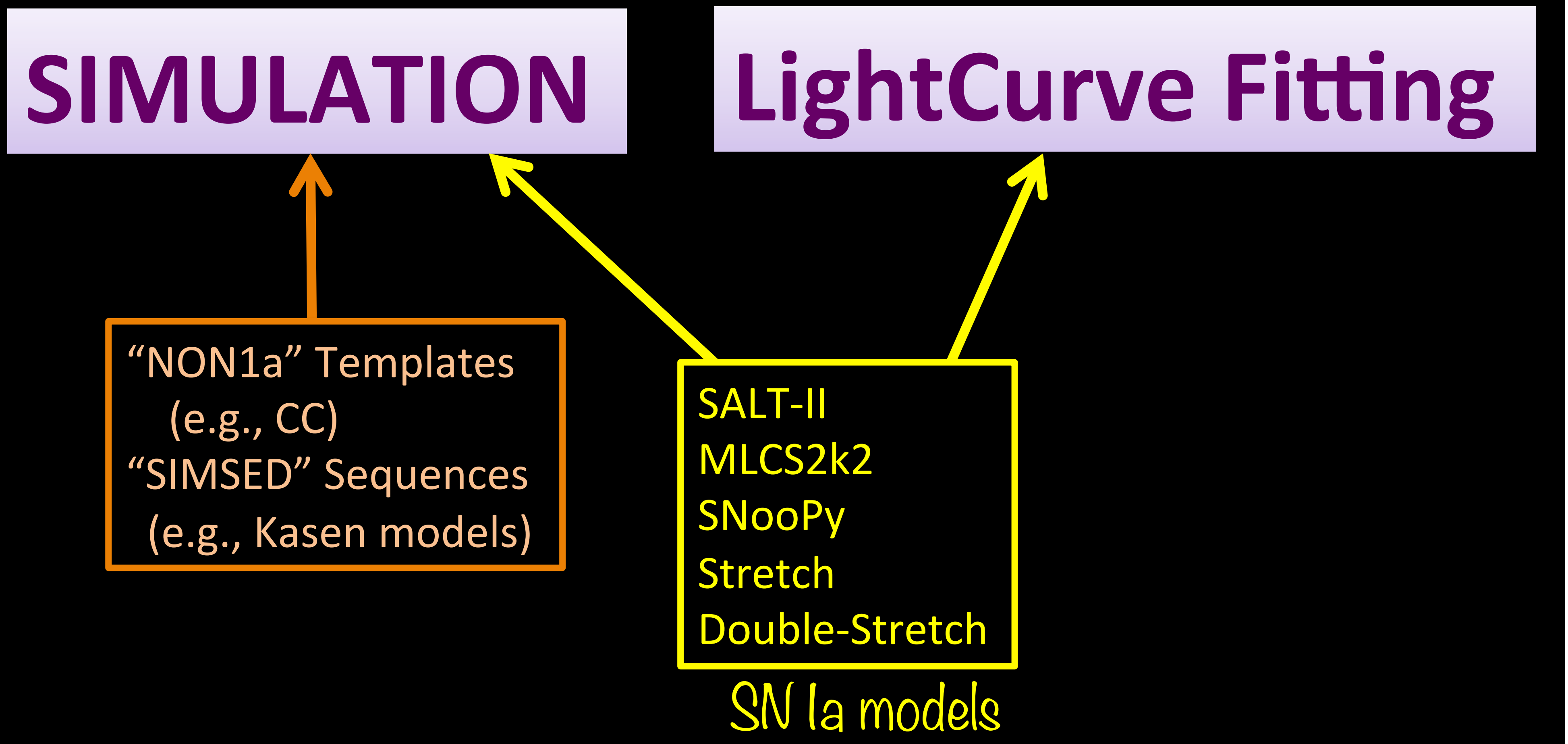
SIMULATION

LightCurve Fitting

SALT-II
MLCS2k2
SNooPy
Stretch
Double-Stretch

SN Ia models

SNANA Architecture



SIMULATION

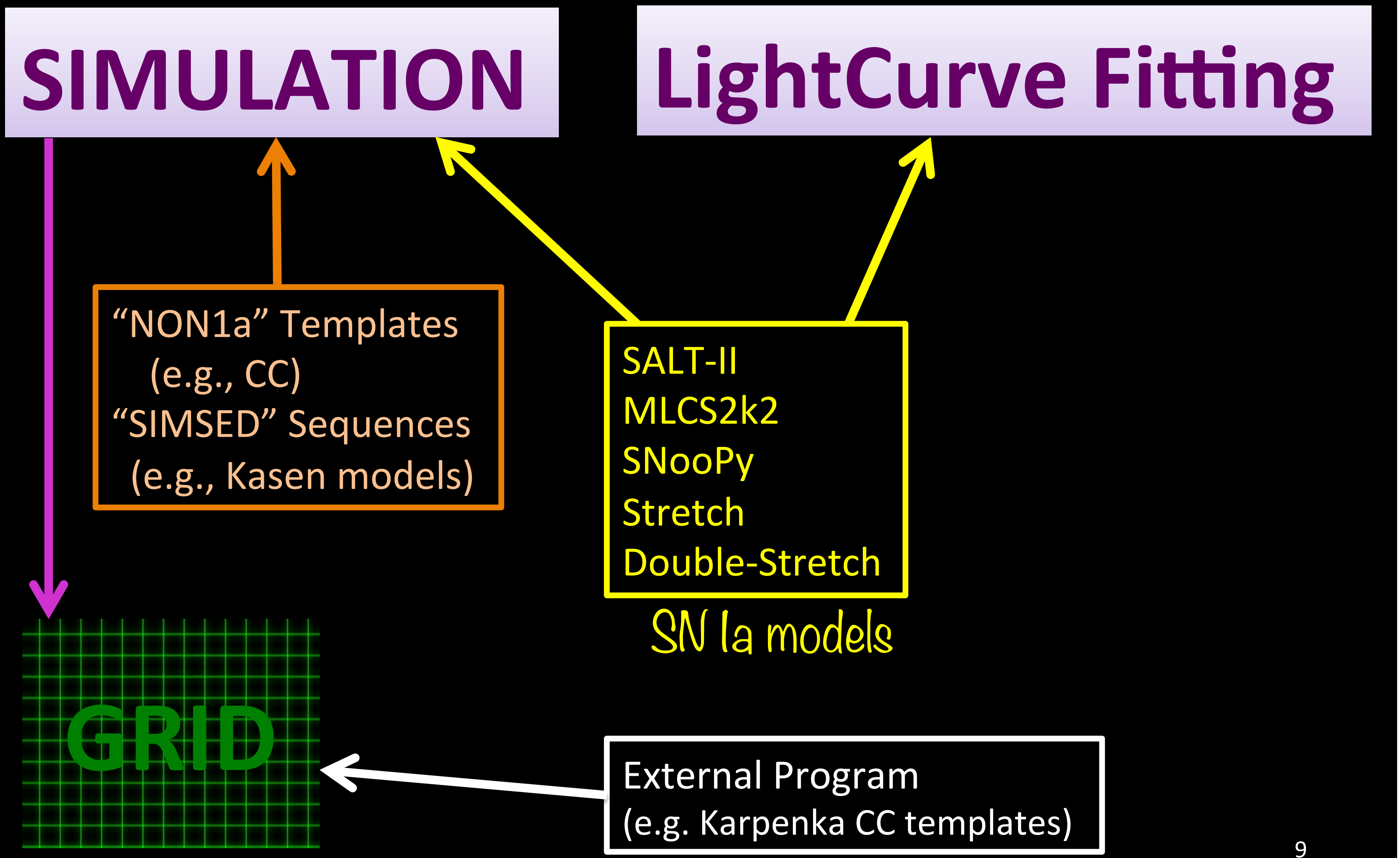
LightCurve Fitting

“NON1a” Templates
(e.g., CC)
“SIMSED” Sequences
(e.g., Kasen models)

SALT-II
MLCS2k2
SNooPy
Stretch
Double-Stretch

SN Ia models

SNANA Architecture



SIMULATION

LightCurve Fitting

"NON1a" Templates
(e.g., CC)
"SIMSED" Sequences
(e.g., Kasen models)

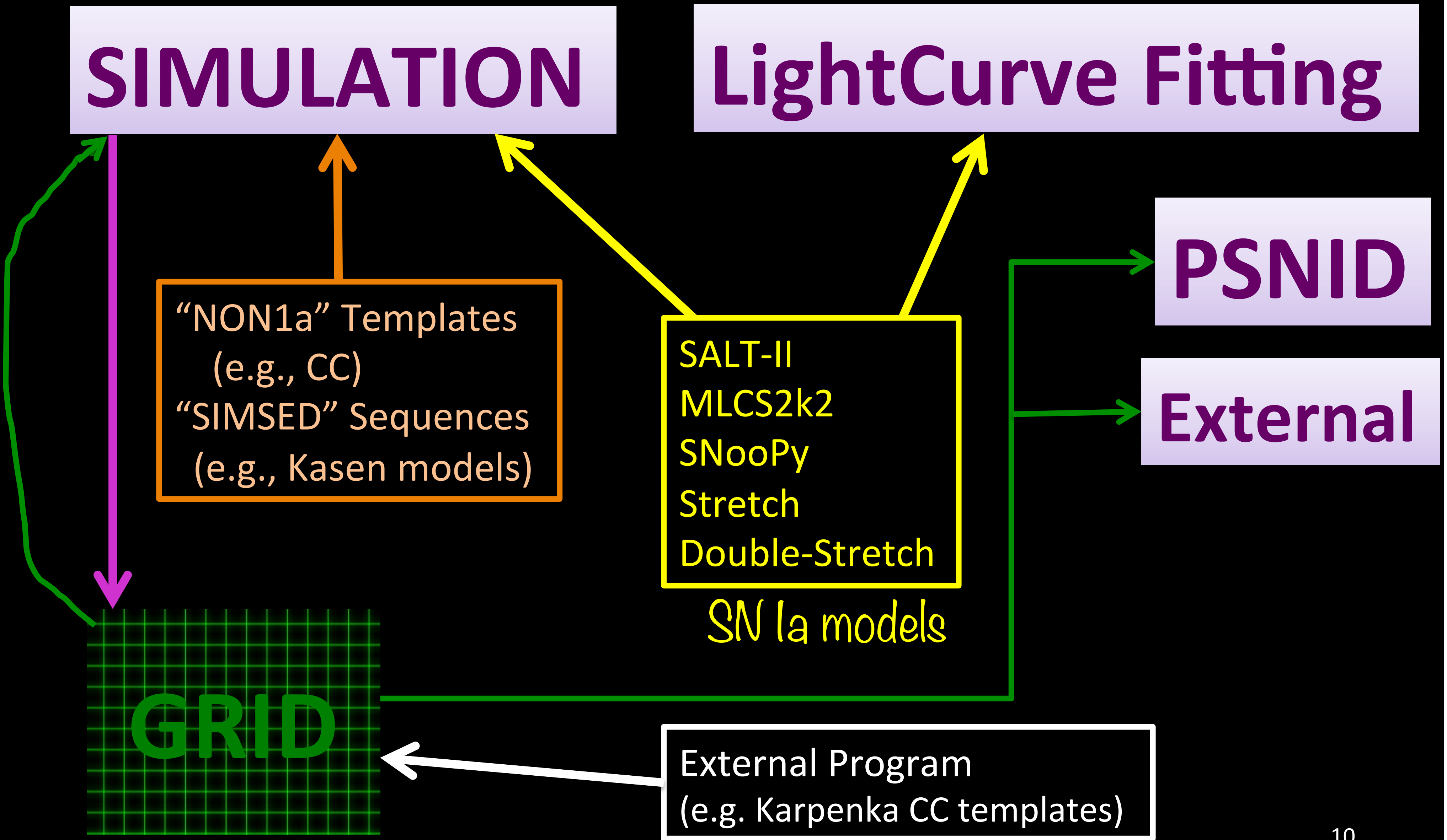
SALT-II
MLCS2k2
SNooPy
Stretch
Double-Stretch

SN Ia models

GRID

External Program
(e.g. Karpenka CC templates)

SNANA Architecture



SIMULATION

LightCurve Fitting

PSNID

External

“NON1a” Templates
(e.g., CC)
“SIMSED” Sequences
(e.g., Kasen models)

SALT-II
MLCS2k2
SNooPy
Stretch
Double-Stretch

SN Ia models

GRID

External Program
(e.g. Karpenka CC templates)

Why Use GRID ?

(versus model params & redshift)

- Allows using templates constructed from non-SNANA programs
- Any SN model → standard model format for template-fitting programs such as PSNID
- Can be faster (e.g., huge speed-up for SNooPy)

SNANA Architecture: File Sharing

SNANA was implicitly designed to run on a cluster with many users sharing files.

Hey, did you update the filter transmission files ?

SNANA Architecture: File Sharing

SNANA was implicitly designed to run on a cluster with many users sharing files.

$SNDATA_ROOT contains

- Data
- Simulated output
- Filter transmissions
- Primary SEDs
- Ia & CC spectral templates
- MW extinction map
- SN model parameters
- SIMLIB files
- HOSTLIB files
- Efficiency maps
- Etc . . .

Most SNANA inputs are in $SNDATA_ROOT.

Can run tests with file(s) in your private directory, but goal is to share files with ***community*** via $SNDATA_ROOT

SNANA Architecture:
Sharing Proprietary Files

During analysis it is useful to share proprietary SNANA files,

$SNDATA_ROOT/INTERNAL/SDSS
$SNDATA_ROOT/INTERNAL/DES
$SNDATA_ROOT/INTERNAL/LSST

INTERNAL directories are not in SNANA downloads.

ENV can be used as part of any input fileName, e.g.,

```
   set $DES_ROOT = $SNDATA_ROOT/INTERNAL/DES
   SIMLIB_FILE:  $DES_ROOT/simlibs/DES_DIFFIMG.SIMLIB
```

SNANA Simulation Capabilities

PHYSICS

- Multiple SNIa + Intrinsic Scatter models
- CC Templates with arbitrary weight, magOff, scatter
- Peculiar Velocities (Gauss scatter)
- Host Galaxy noise, photo-z, SN correlations (HOSTLIB)
- Galactic Extinction (SFD98 or Schlaffly 2012 update)
- Arbitrary z-dependence for any param: e.g., β(z)

SURVEY

- Use measured cadence, skyNoise, ZP, PSF (SIMLIB)
- Wrong-Host model with incorrect z_{Host}
- Flux-Error Correction (e.g., from fakes on images)
- Survey Detection Effic vs. S/N (feeds trigger logic)
- Trigger Logic (e.g., 2 nights above threshold)
- Survey Effic map for Spec-Confirmed
- Survey Effic map for Spec-z_{Host} (e.g., OzDES effic)
- Survey Effic map for photo-z_{Host} (e.g., for SN+host photo-z fit)

SNANA Simulation Capabilities

- Multiple SNIa + Intrinsic Scatter models
- CC Templates with arbitrary weight, magOff, scatter
- Peculiar Velocities (Gauss scatter)
- Host Galaxy noise, photo-z, SN correlations (HOSTLIB)
- Galactic Extinction (SFD98 or Schlaffly 2012 update)
- Arbitrary z-dependence for any param: e.g., β(z)

PHYSICS

- Use measured cadence, skyNoise, ZP, PSF (SIMLIB)
- Wrong-Host model with incorrect z_{Host}

SURVEY

Characterizing the survey is crucial to avoid mis-interpreting instrumental artifacts as astrophysics.

to-z fit)

SNANA CPU Proc-Time

Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz

- DES Simulation
  - Generate SALT-II light curves: 70 Hz
    → accepted rate: 8 Hz
  - Generate CC light curves: 600 Hz
    → accepted rate: 5 Hz
- Light Curve Fitting (DES) with SALT-II: 4 Hz

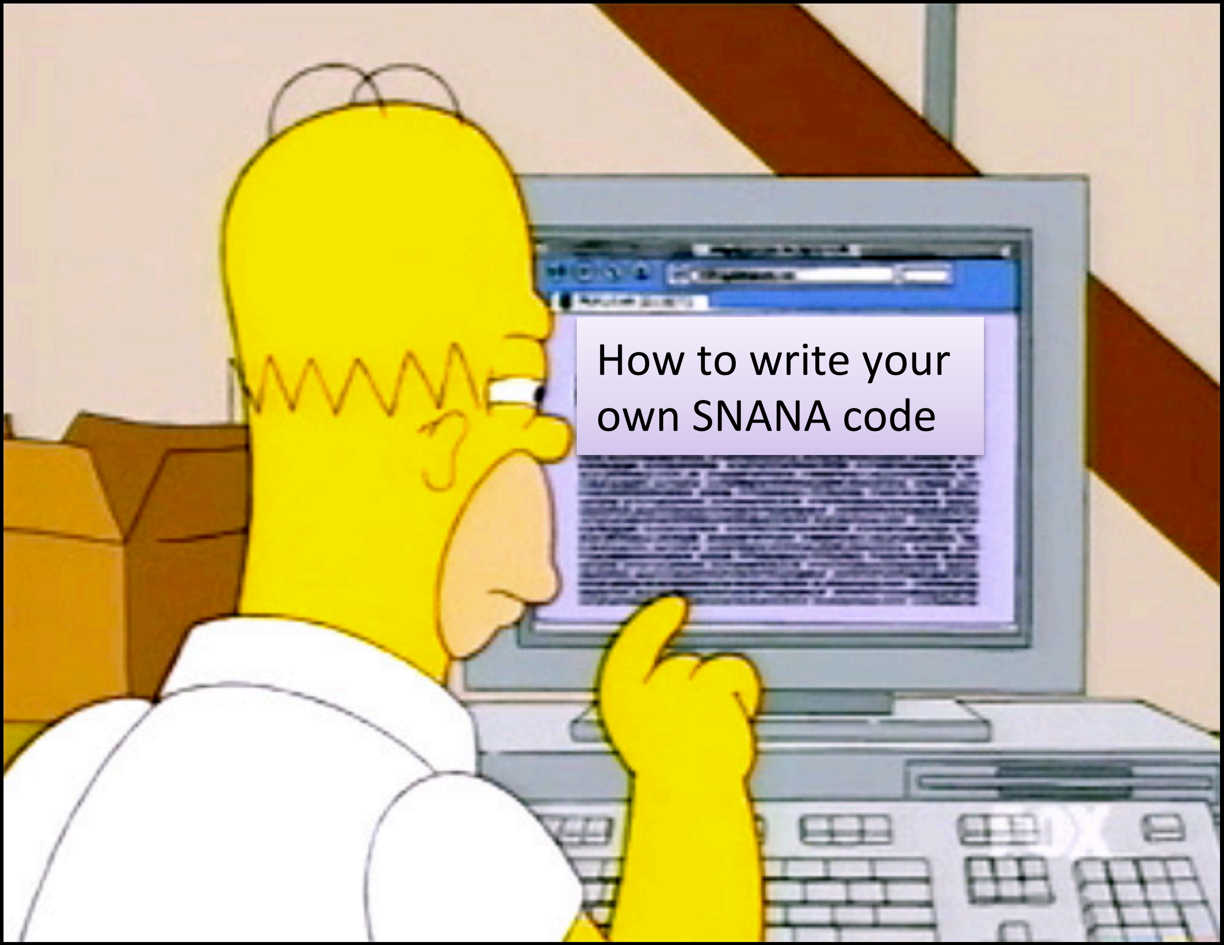
- Note that Processing time scales with number of observations and redshift range.

Simulation Speed-up Trick

When efficiency is very low (e.g., for CC), generation speed can be limited by reading a new SIMLIB entry for each event.

Trick: use "SIMLIB_NREPEAT" key to re-use each SIMLIB entry many times before reading the next one.

Software Interface to Add Private Code



No Interface for Simulation

- However, without re-compiling can add
  + new survey, filters, SIMLIB, HOSTLIB,
  calibration info
  + new CC templates, weights
  + new SIMSED model (e.g., Kilonova)

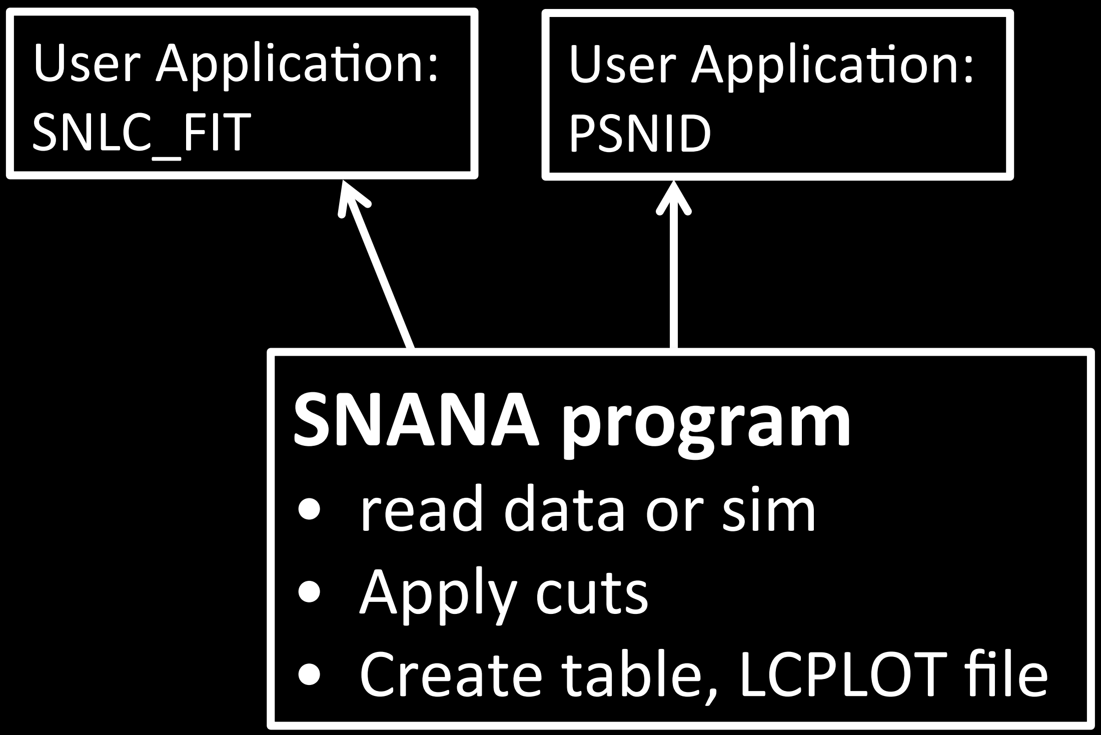
(new “software model” takes ~hour to install)

Interface for Analysis

SNANA program

- read data or sim
- Apply cuts
- Create table, LCPLOT file

Interface for Analysis



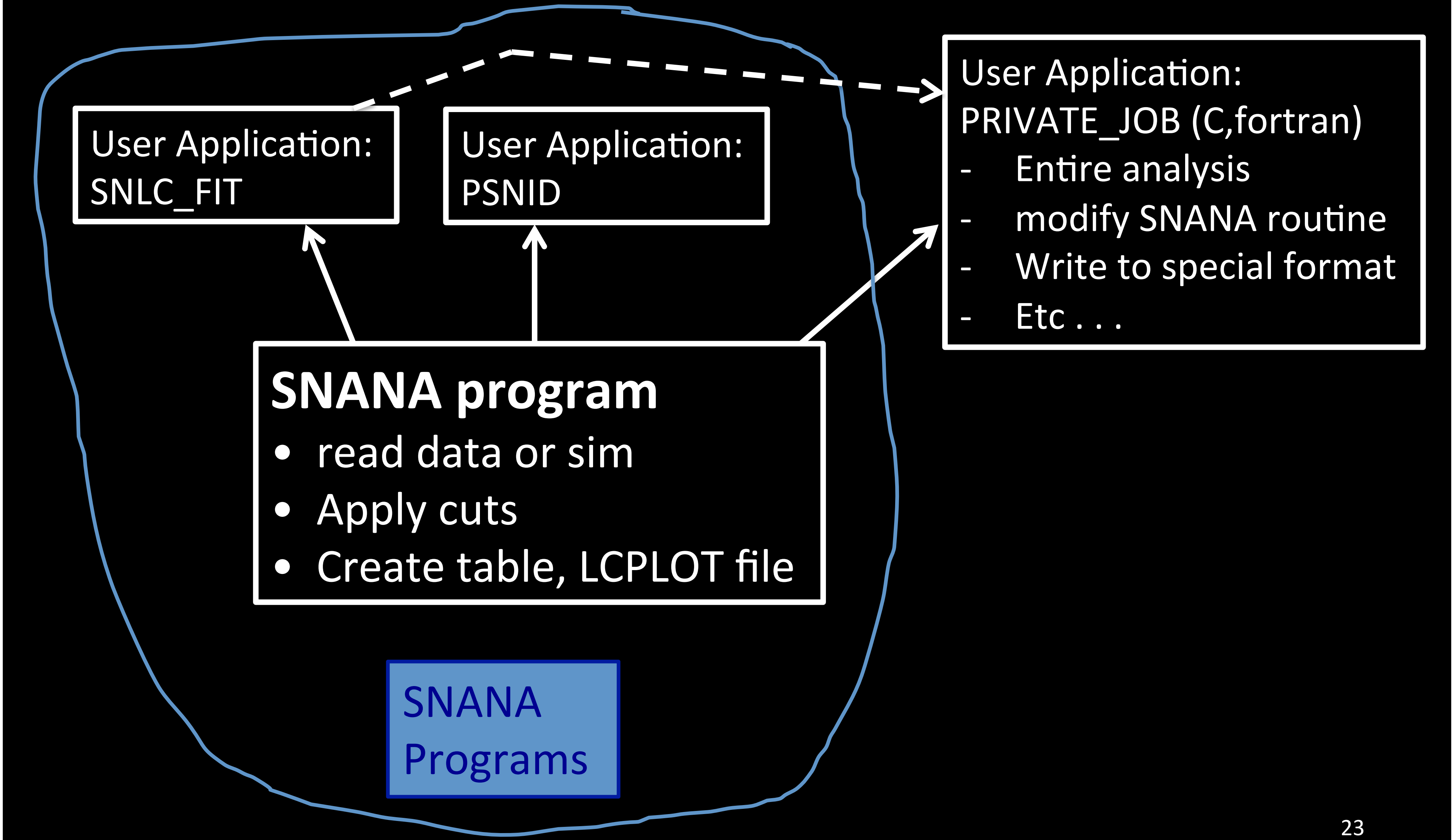
User Application:
SNLC_FIT

User Application:
PSNID

SNANA program

- read data or sim
- Apply cuts
- Create table, LCPLOT file

Interface for Analysis



User Application:
SNLC_FIT

User Application:
PSNID

User Application:
PRIVATE_JOB (C,fortran)

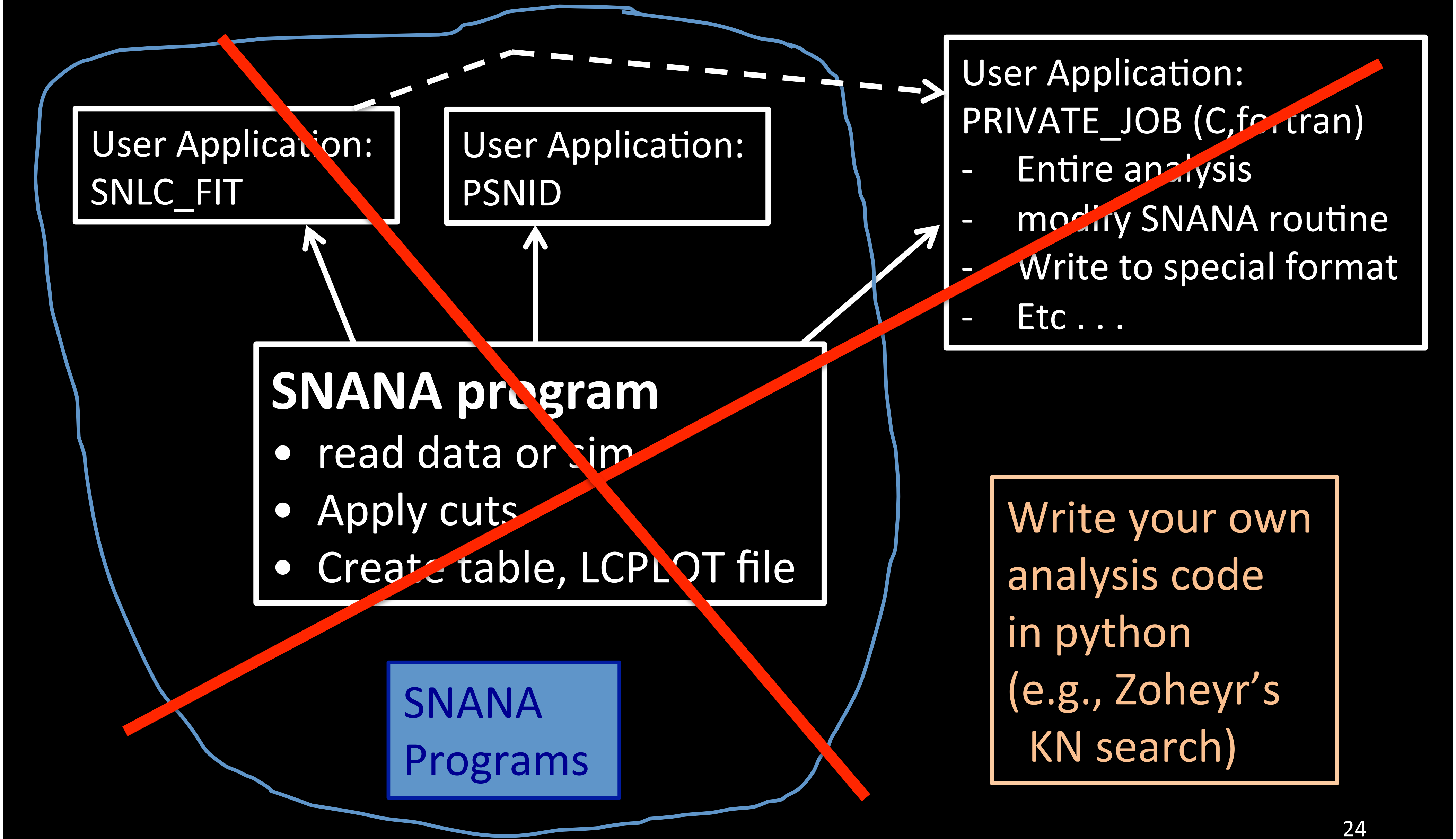
- Entire analysis
- modify SNANA routine
- Write to special format
- Etc . . .

SNANA program

- read data or sim
- Apply cuts
- Create table, LCPLOT file

SNANA
Programs

Interface for Analysis



User Application:
SNLC_FIT

User Application:
PSNID

User Application:
PRIVATE_JOB (C,fortran)
- Entire analysis
- modify SNANA routine
- Write to special format
- Etc . . .

SNANA program
- read data or sim
- Apply cuts
- Create table, LCPLOT file

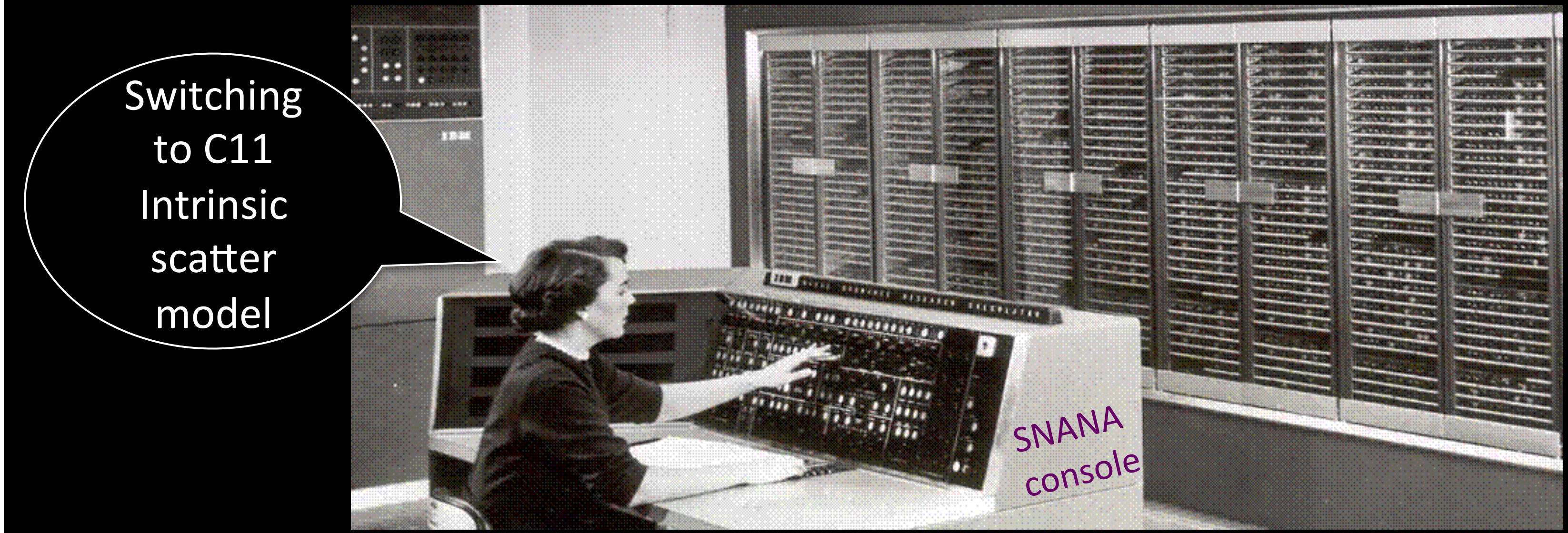
SNANA
Programs

Write your own
analysis code
in python
(e.g., Zoheyr's
KN search)

Systematics & Multi-Core Jobs

- Precision analyses typically require many iterations of simulations and analysis, each with a small variation in parameters or method.
- SNANA has tools to implement multi-iteration analyses using multi-core platforms.

(e.g., Fermilab, NERSC, Argonne, Midway, Folio . . .)



Systematics & Multi-Core Jobs

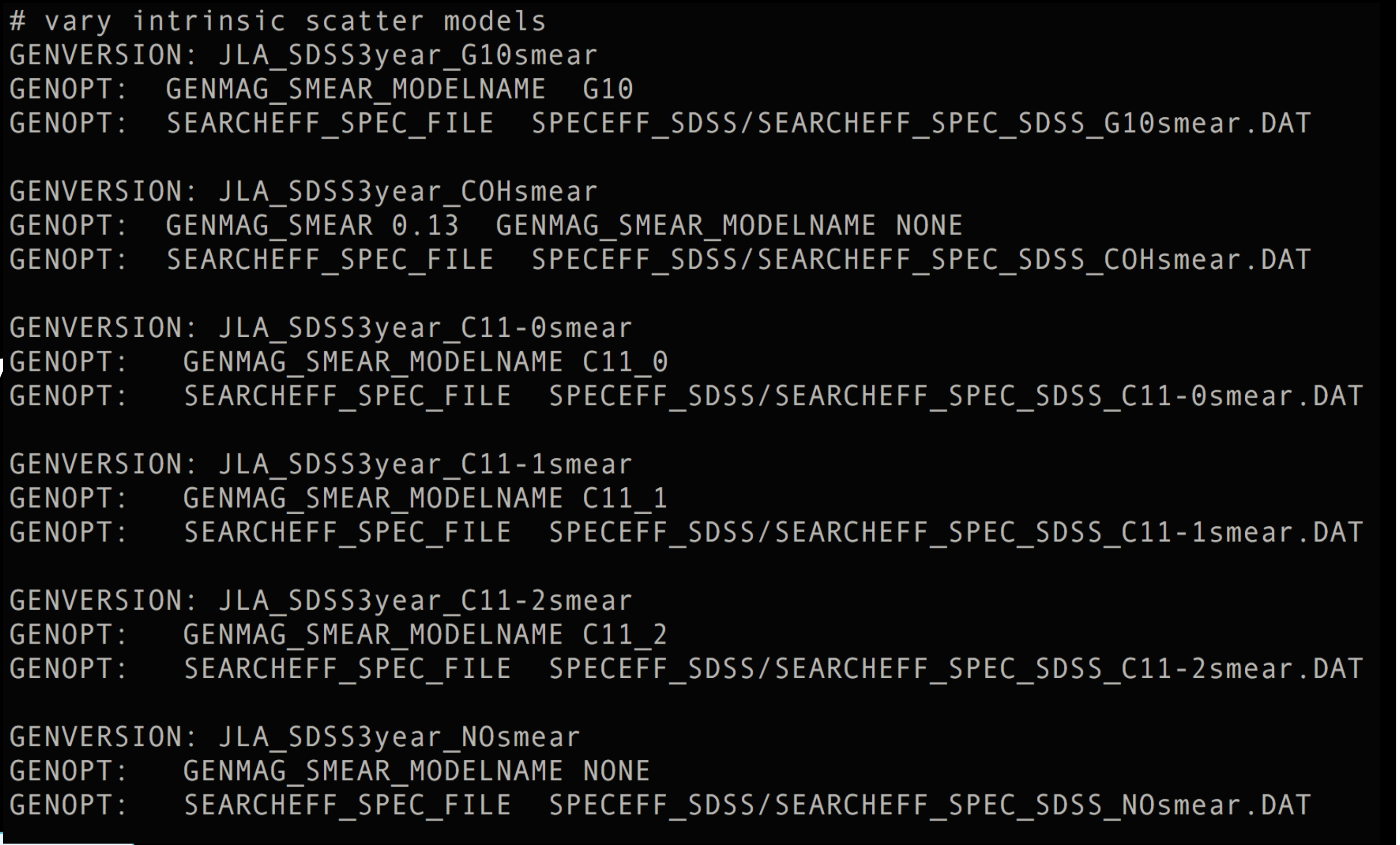
- Sim & Analysis codes read input instructions from a text file.
- Specify variations with **command-line overrides** to avoid more input files.
- SNANA Scripts use **command-line override** feature to launch multiple jobs in batch system.

sim_SNmix.pl
Launch multiple
SIM(Ia+CC) jobs

split_and_fit.pl
Launch multiple
lightcurve fit jobs

Systematics & Multi-Core Jobs

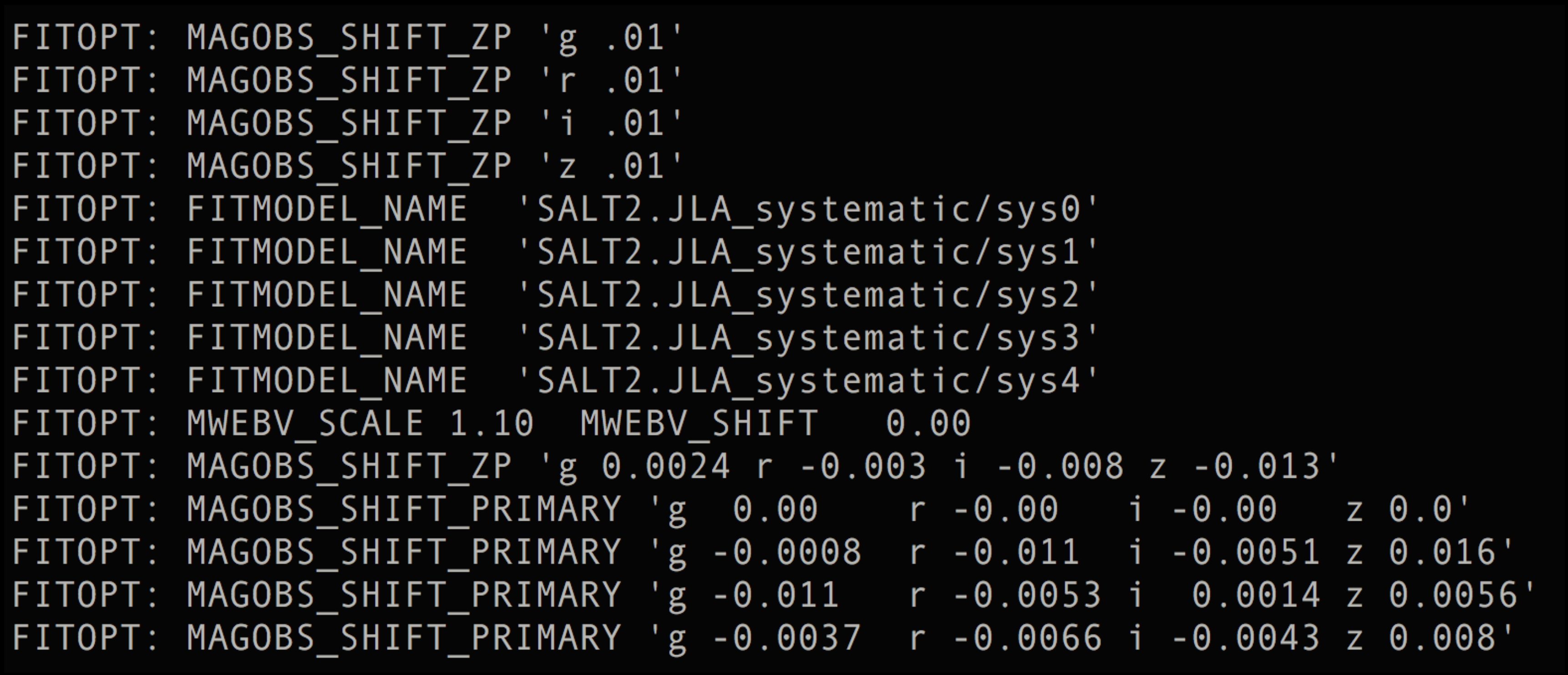
Example:
Subset of sim jobs for JLA systematics.
Each job → separate core



```
# vary intrinsic scatter models
GENVERSION: JLA_SDSS3year_G10smear
GENOPT:  GENMAG_SMEAR_MODELNAME  G10
GENOPT:  SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_G10smear.DAT

GENVERSION: JLA_SDSS3year_COHsmear
GENOPT:  GENMAG_SMEAR 0.13  GENMAG_SMEAR_MODELNAME NONE
GENOPT:  SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_COHsmear.DAT

GENVERSION: JLA_SDSS3year_C11-0smear
GENOPT:   GENMAG_SMEAR_MODELNAME C11_0
GENOPT:   SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_C11-0smear.DAT

GENVERSION: JLA_SDSS3year_C11-1smear
GENOPT:   GENMAG_SMEAR_MODELNAME C11_1
GENOPT:   SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_C11-1smear.DAT

GENVERSION: JLA_SDSS3year_C11-2smear
GENOPT:   GENMAG_SMEAR_MODELNAME C11_2
GENOPT:   SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_C11-2smear.DAT

GENVERSION: JLA_SDSS3year_NOsmear
GENOPT:   GENMAG_SMEAR_MODELNAME NONE
GENOPT:   SEARCHEFF_SPEC_FILE  SPECEFF_SDSS/SEARCHEFF_SPEC_SDSS_NOsmear.DAT
```

sim_SNmix.pl
Launch multiple SIM(Ia+CC) jobs

Systematics & Multi-Core Jobs



```
FITOPT: MAGOBS_SHIFT_ZP 'g .01'
FITOPT: MAGOBS_SHIFT_ZP 'r .01'
FITOPT: MAGOBS_SHIFT_ZP 'i .01'
FITOPT: MAGOBS_SHIFT_ZP 'z .01'
FITOPT: FITMODEL_NAME  'SALT2.JLA_systematic/sys0'
FITOPT: FITMODEL_NAME  'SALT2.JLA_systematic/sys1'
FITOPT: FITMODEL_NAME  'SALT2.JLA_systematic/sys2'
FITOPT: FITMODEL_NAME  'SALT2.JLA_systematic/sys3'
FITOPT: FITMODEL_NAME  'SALT2.JLA_systematic/sys4'
FITOPT: MWEBV_SCALE 1.10  MWEBV_SHIFT   0.00
FITOPT: MAGOBS_SHIFT_ZP 'g 0.0024 r -0.003 i -0.008 z -0.013'
FITOPT: MAGOBS_SHIFT_PRIMARY 'g  0.00    r -0.00   i -0.00   z 0.0'
FITOPT: MAGOBS_SHIFT_PRIMARY 'g -0.0008  r -0.011  i -0.0051 z 0.016'
FITOPT: MAGOBS_SHIFT_PRIMARY 'g -0.011   r -0.0053 i  0.0014 z 0.0056'
FITOPT: MAGOBS_SHIFT_PRIMARY 'g -0.0037  r -0.0066 i -0.0043 z 0.008'
```

Example:
Subset of fit jobs for PS1 systematics

split_and_fit.pl
Launch multiple lightcurve fit jobs

SNANA Output

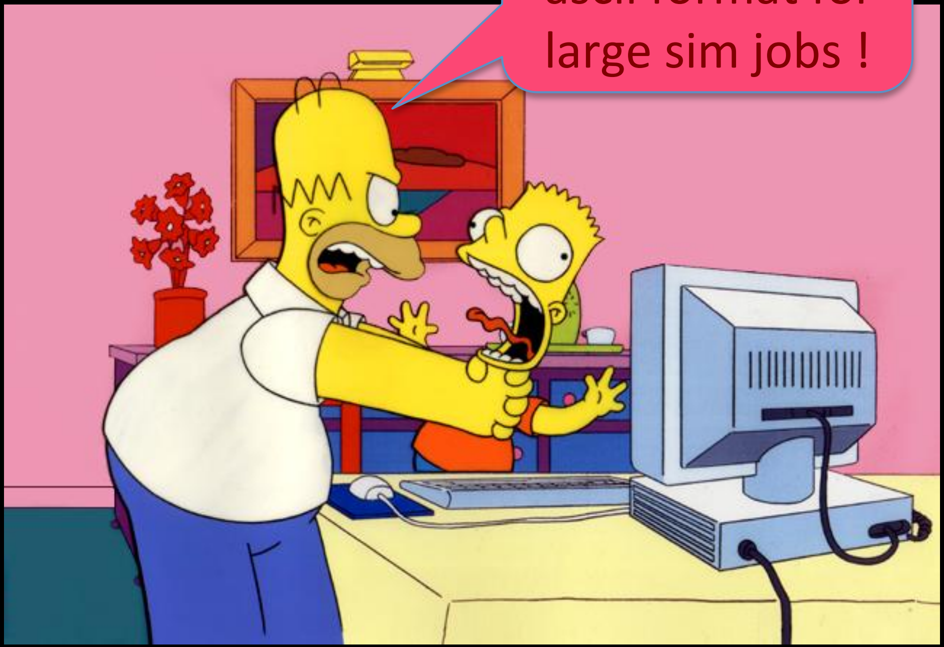
SIMULATION Output

Data Files

- FITS format for large jobs

or

- ASCII format for testing (1 file per SN)



I said NEVER use ascii format for large sim jobs !

SIMULATION Output

Data Files

- FITS format for large jobs

or

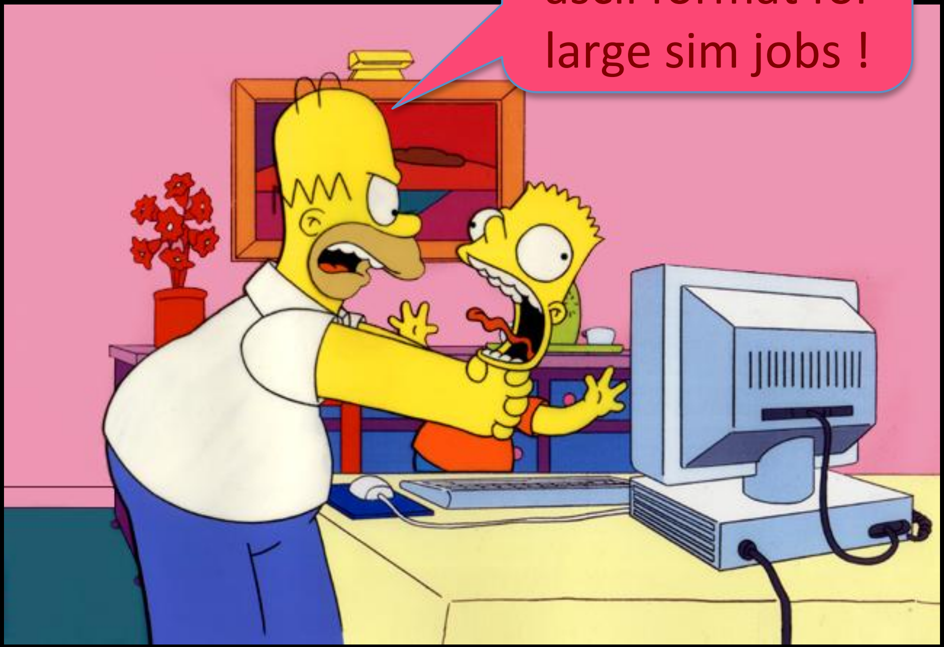
- ASCII format for testing
  (1 file per SN)

ASCII Summary File

- 1 row per accepted SN

or

- 1 row per generated SN
  (for efficiency)



I said NEVER use ascii format for large sim jobs !

SIMULATION Output

Never Trust Simulation Output

→

Always check data/MC distributions (redshift, fit params, SNR, etc . . .)

Y'all don't have to worry none; I generated them simulations myself.
Who took my beer ?

Analysis Output → SNTABLEs

- SNANA table before fit (1 row per SN)
- FITRES table after fit (1 row per SN)
- LCPLOT table with light curve & best-fit curve

Analysis Output → SNTABLEs

- SNANA table before fit (1 row per SN)
- FITRES table after fit (1 row per SN)
- LCPLOT table with light curve & best-fit curve

Formats:

- TEXT (1 file per table)
- HBOOK (all tables → 1 file)
- ROOT (all tables → 1 file)

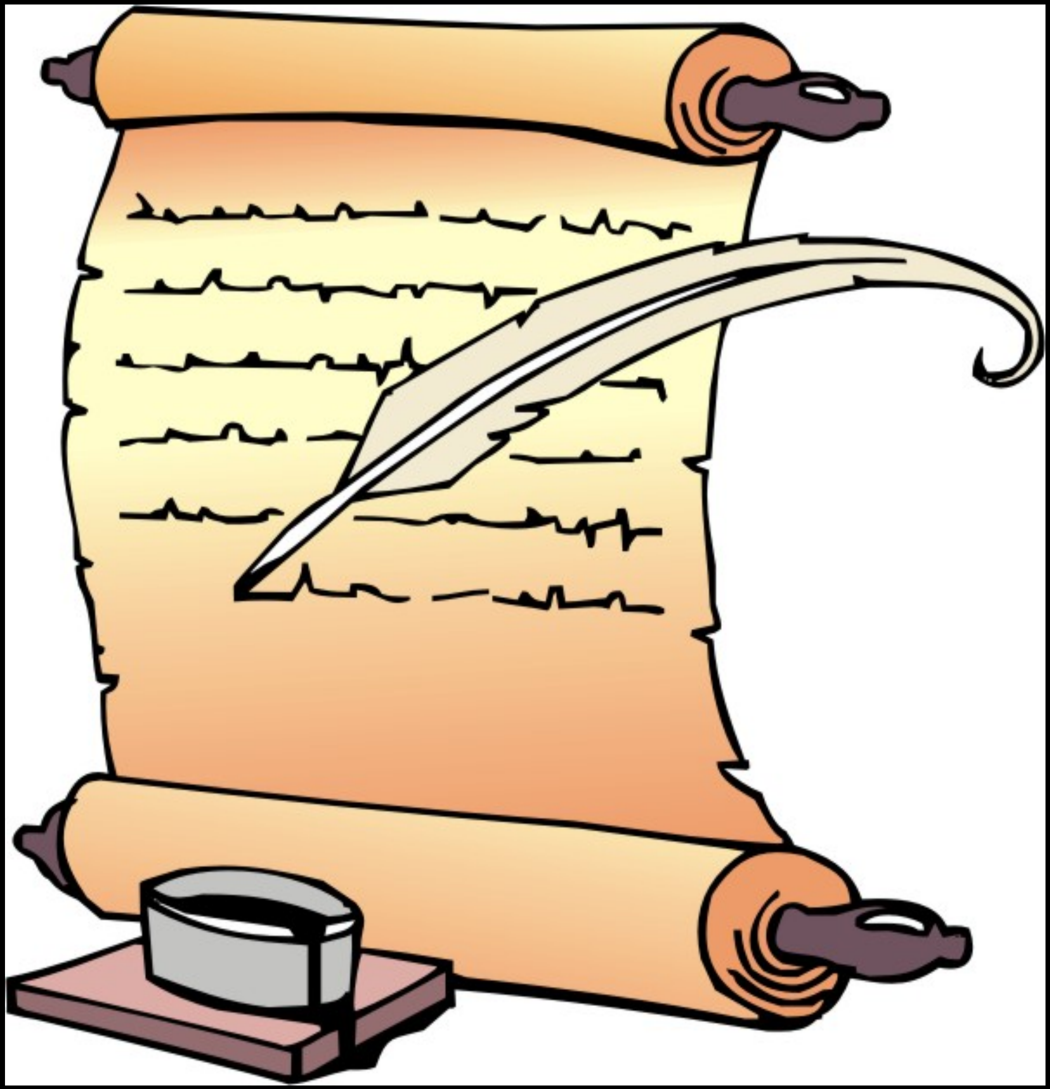
- Easier to install SNANA without HBOOK or ROOT (but will miss most output variables from analysis)
- New format can be added if interface routines are provided

Analysis Output → SNTABLEs

- SNANA table before fit (1 row per SN)
- FITRES table after fit (1 row per SN)

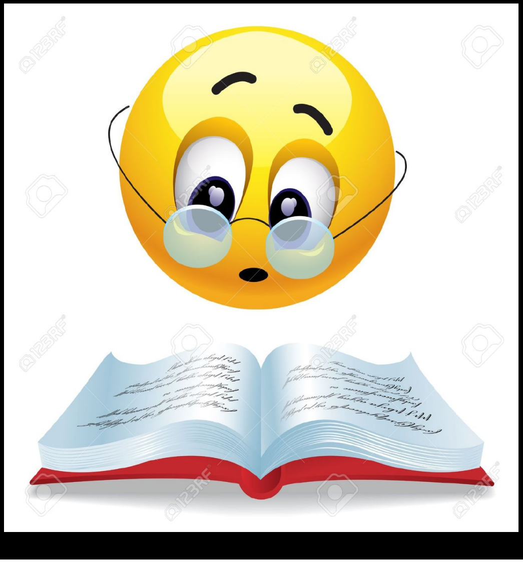
- ALL variables stored in **HBOOK** & **ROOT** format
  (compared to TEXT, more efficient to read & write)
- Subset stored in **TEXT** for input to cosmology fit
- Utility to append **TEXT** file (from HBOOK or ROOT)
- See "sntable_dump" to view, dump, append tables
- Slowly moving away **TEXT** (except for debug)

SNANA Documentation

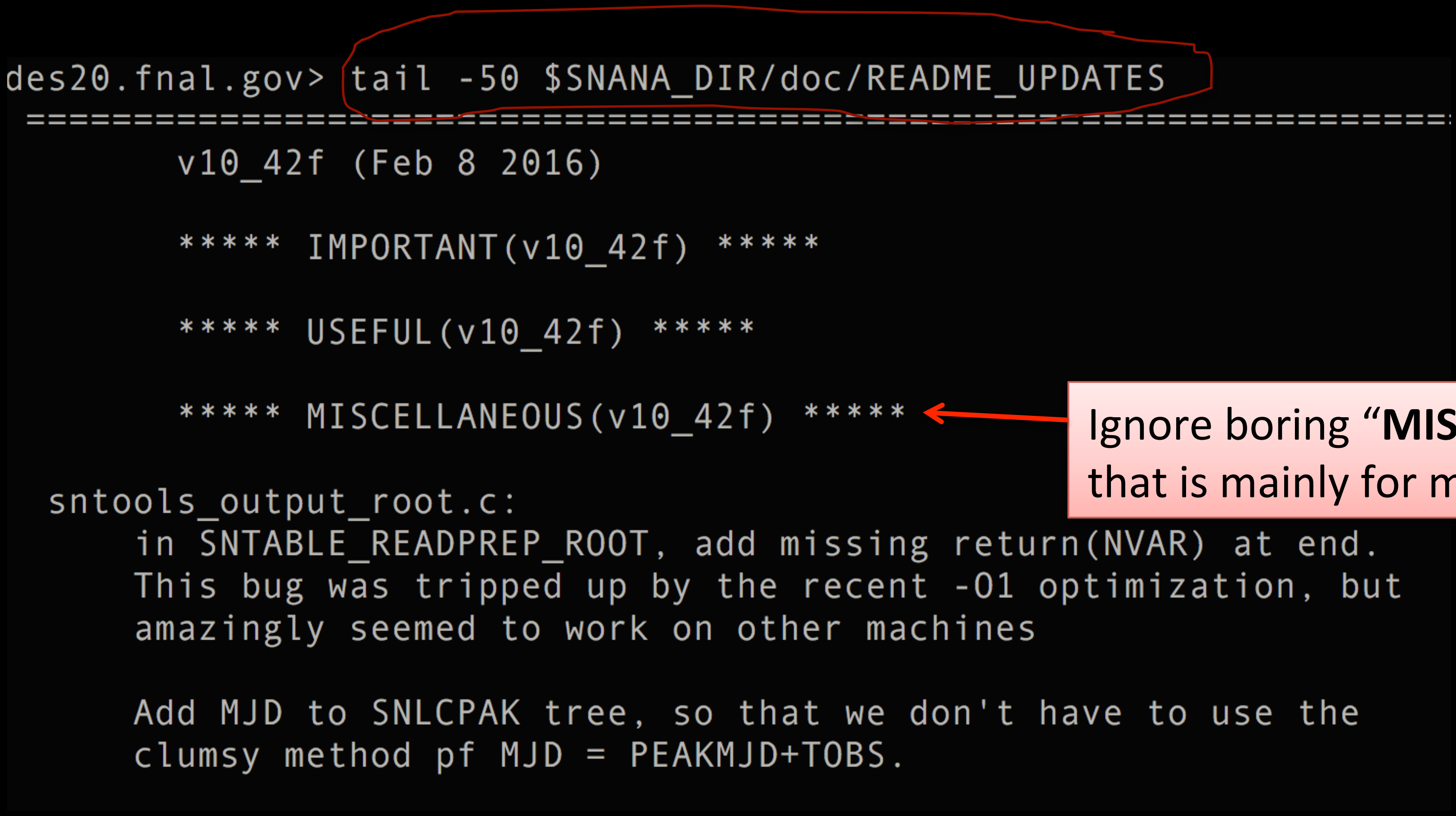


User Interface for Manual

- 120 page manual with no interface except ‘preview’
- Difficult to get started without using examples from somebody else.
- Would appreciate community help making this better.



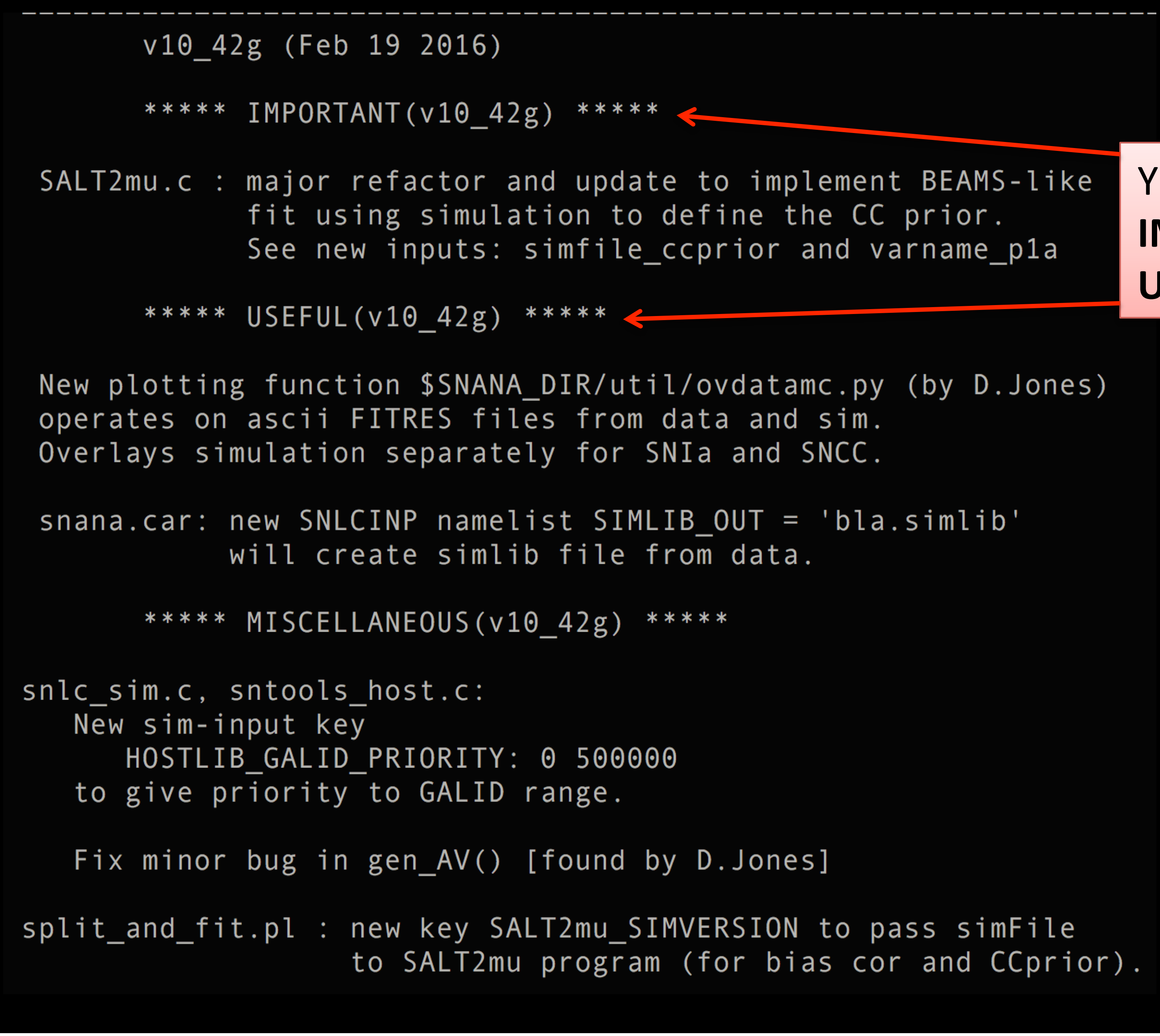
Tracking Changes



```
des20.fnal.gov> tail -50 $SNANA_DIR/doc/README_UPDATES
 ===================================================================
        v10_42f (Feb 8 2016)

        ***** IMPORTANT(v10_42f) *****

        ***** USEFUL(v10_42f) *****

        ***** MISCELLANEOUS(v10_42f) *****

  sntools_output_root.c:
      in SNTABLE_READPREP_ROOT, add missing return(NVAR) at end.
      This bug was tripped up by the recent -O1 optimization, but
      amazingly seemed to work on other machines

      Add MJD to SNLCPAK tree, so that we don't have to use the
      clumsy method pf MJD = PEAKMJD+TOBS.
```

Ignore boring "**MISC**" that is mainly for me

Tracking Changes



```
------------------------------------------------------------------
       v10_42g (Feb 19 2016)

       ***** IMPORTANT(v10_42g) *****

 SALT2mu.c : major refactor and update to implement BEAMS-like
             fit using simulation to define the CC prior.
             See new inputs: simfile_ccprior and varname_p1a

       ***** USEFUL(v10_42g) *****

 New plotting function $SNANA_DIR/util/ovdatamc.py (by D.Jones)
 operates on ascii FITRES files from data and sim.
 Overlays simulation separately for SNIa and SNCC.

 snana.car: new SNLCINP namelist SIMLIB_OUT = 'bla.simlib'
            will create simlib file from data.

       ***** MISCELLANEOUS(v10_42g) *****

snlc_sim.c, sntools_host.c:
   New sim-input key
      HOSTLIB_GALID_PRIORITY: 0 500000
   to give priority to GALID range.

   Fix minor bug in gen_AV() [found by D.Jones]

split_and_fit.pl : new key SALT2mu_SIMVERSION to pass simFile
                   to SALT2mu program (for bias cor and CCprior).
```

You should read **IMPORTANT** & **USEFUL** updates

Miscellaneous

- Report bugs ASAP; don't just hack a private fix for yourself.
- Report compilation warnings.
- Think about how you can contribute to SNANA.
- In papers, SNANA citation is not enough; also cite source of models, template data, galaxy catalogs, etc.

Conclusion

From the SNANA Legal Team:

- **SIDE EFFECTS** include, but are not limited to: confusion, frustration, watery eyes, head-aches, weight loss, weight gain, systematics-limited results, incorrect results, denial of tenure.
- Do not drive or operate heavy machinery while using SNANA.