Back to the Future: Ideas for SN Data Model

Rick

June 4 2021

Preface

- This is a PREQUEL to the DC2 SNIa-analysis tutorial;
  $SNANA_LSST_ROOT/starterKits/DC2+SNANA on Cori
- Tutorial began with **data files** nicely prepared by Bruno
- Here we examine
  - how/why **data files** are created
  - Challenges & ideas to create **data files** in LSST-SN
- Ability to rapidly and reliably create **data files** will greatly impact duration and quality of first cosmology analysis.

Unresolved issues are indicated with confused emoji

How should we access SN data for analysis ?

- **Interrogate Data base** :
  - difficult for existing codes that read data files
  - slower analysis
  - difficult to make public
  - ingest older data sets into DB ?

- **data files created from DB extraction tool**
  - only one code needs to access data base
  - existing analysis codes process LSST data in the same was as older data
  - easy to make public
  - multiple uses: SNIa-cosmology, SN physics, TVS

What data format ?

- Following DC2-SNIa tutorial, consider SNANA FITS format
- Design criteria (from many years ago)
  - scales to millions of events (e.g. 3M for PLASTICC)
  - fast access, even if searching for last event (uses pointers to quickly find photometry)
  - same code writes data for real and sim events
  - portable among different computer platforms
  - variable storage size based on need: short int, int, float, double

Contents of 3 FITS data files

- **HEADER** Info:
  - **measured**: zHEL[ERR] RA DEC TYPE HOST m_{HOST} m_{SB}
  - **computed**: SNID IAUC zCMB[ERR] VPEC[ERR] $t0_{approx}$ MWEBV[ERR] HOST[sep DLR mass SFR, z_{PHOT}]
- **OBSERVATION** info:
  - **measured**: MJD, FLUXCAL[ERR]
  - **camera**: FIELD BAND CCDNUM
  - **photom status**: PHOTFLAG PHOTPROB
  - **image properties**: ZP PSF σ_{SKY} σ_{tSKY} XPIX YPIX
- **SPECTRA** info (e.g., for model training):
  - **measured**: MJD, wave, dF/dLam[ERR]
  - **computed**: SN-phase

- Here is an example data folder for DES

```
DESALL_forcePhoto_real_snana_fits.IGNORE
DESALL_forcePhoto_real_snana_fits.LIST
DESALL_forcePhoto_real_snana_fits.README
DESY1_forcePhoto_real_snana_fits_HEAD.FITS
DESY1_forcePhoto_real_snana_fits_PHOT.FITS
DESY2_forcePhoto_real_snana_fits_HEAD.FITS
DESY2_forcePhoto_real_snana_fits_PHOT.FITS
DESY3_forcePhoto_real_snana_fits_HEAD.FITS
DESY3_forcePhoto_real_snana_fits_PHOT.FITS
DESY4_forcePhoto_real_snana_fits_HEAD.FITS
DESY4_forcePhoto_real_snana_fits_PHOT.FITS
DESY5_forcePhoto_real_snana_fits_HEAD.FITS
DESY5_forcePhoto_real_snana_fits_PHOT.FITS
```

HEAD files:
info per event

PHOT files:
info per OBS

- Here is an example data folder for DES

```
DESALL_forcePhoto_real_snana_fits.IGNORE
DESALL_forcePhoto_real_snana_fits.LIST
DESALL_forcePhoto_real_snana_fits.README
DESY1_forcePhoto_real_snana_fits_HEAD.FITS
DESY1_forcePhoto_real_snana_fits_PHOT.FITS
DESY2_forcePhoto_real_snana_fits_HEAD.FITS
DESY2_forcePhoto_real_snana_fits_PHOT.FITS
DESY3_forcePhoto_real_snana_fits_HEAD.FITS
DESY3_forcePhoto_real_snana_fits_PHOT.FITS
DESY4_forcePhoto_real_snana_fits_HEAD.FITS
DESY4_forcePhoto_real_snana_fits_PHOT.FITS
DESY5_forcePhoto_real_snana_fits_HEAD.FITS
DESY5_forcePhoto_real_snana_fits_PHOT.FITS
```

season and field are useful to organize DDF …

but not clear how to organize WFD ?

- Here is an example data folder for DES fakes:

```
DESALL_forcePhoto_fake_snana_fits.IGNORE
DESALL_forcePhoto_fake_snana_fits.LIST
DESALL_forcePhoto_fake_snana_fits.README
DESY1_forcePhoto_fake_snana_fits_HEAD.FITS
DESY1_forcePhoto_fake_snana_fits_PHOT.FITS
DESY2_forcePhoto_fake_snana_fits_HEAD.FITS
DESY2_forcePhoto_fake_snana_fits_PHOT.FITS
DESY3_forcePhoto_fake_snana_fits_HEAD.FITS
DESY3_forcePhoto_fake_snana_fits_PHOT.FITS
DESY4_forcePhoto_fake_snana_fits_HEAD.FITS
DESY4_forcePhoto_fake_snana_fits_PHOT.FITS
DESY5_forcePhoto_fake_snana_fits_HEAD.FITS
DESY5_forcePhoto_fake_snana_fits_PHOT.FITS
FAKES_OVERLAID_AllFakes_v3.DAT
```

Same data structure for fakes overlaid on images

TEXT Format Alternative

- small data sets use TEXT format (one file per SN)
- While FITS is convenient for large data sets, it's difficult for visual debugging; thus SNANA has extraction tool for FITS → TEXT (up to 100 events)

```
SURVEY:     LSST
FAKE:       1
SNID:       MS_9300_3885
FILTERS:    ugrizY
PIXSIZE:    0.199598 # arcsecs
NXPIX:      4072
NYPIX:      4000
RA:         55.579078 # deg
DEC:        -31.916440 # deg
MWEBV:      0 +- 0
REDSHIFT_FINAL:  0.9575 +- 0.0001  # CMB
HOSTGAL_OBJID:  12345

PRIVATE(SIM_PEAKMJD):  60272.1552
PRIVATE(SIM_SALT2x0):      1.9488e-06
PRIVATE(SIM_SALT2mB):     24.9056
PRIVATE(SIM_SALT2c) :      0.0096
PRIVATE(SIM_SALT2x1):      0.9746
PRIVATE(SIM_HOSTLESS):   False
PRIVATE(NOBS):    34
PRIVATE(TRESTMIN):   -19.4192
PRIVATE(TRESTMAX):    49.4890

# ================================
NOBS:  34
NVAR:  14
VARLIST:   MJD FLT FIELD PHOTFLAG     XPIX     YPIX CCDNUM GAIN    FLUXCAL FLUXCALERR ZPFLUX    NEA SKYSIG SIM_MAGOBS
OBS: 60234.1419 r    WFD        0  1649.72  3652.32    153 0.70  7.088e-01  3.618e+00 32.176  41.50  38.04 33.180
OBS: 60234.1560 r    WFD        0   303.57   965.98    186 0.70 -3.595e+00  3.554e+00 32.176  40.17  37.89 33.150
OBS: 60234.1583 r    WFD        0   164.12  2121.87      2 0.70 -5.109e+00  3.396e+00 32.176  36.33  37.67 33.146
OBS: 60240.1353 r    WFD        0  2009.16  3806.47    101 0.70  5.187e+00  6.327e+00 32.168  55.05  61.89 28.981
OBS: 60240.2046 r    WFD        0   347.55  1926.49    139 0.70  9.735e+00  5.837e+00 32.169  55.53  55.41 28.958
OBS: 60241.2161 i    WFD        0  3637.58  1524.76    187 0.70 -1.158e+01  7.680e+00 31.849  40.17  63.20 28.308
OBS: 60241.2327 i    WFD        0  1567.77   657.14    180 0.70  5.627e+00  7.906e+00 31.847  42.74  63.11 28.298
OBS: 60242.1090 Y    WFD        0  2490.19  1480.20    150 0.70 -5.662e+00  3.294e+01 30.633  73.86  67.09 26.954
OBS: 60242.1220 Y    WFD        0  2569.98  1823.81     23 0.70 -4.592e+01  3.191e+01 30.634  64.65  68.69 26.949
OBS: 60242.1225 Y    WFD        0  1347.04   820.19    140 0.70  2.246e+01  3.236e+01 30.635  65.68  68.88 26.949
OBS: 60242.2202 i    WFD        0   588.05  3765.95    102 0.70  1.558e+01  7.209e+00 31.852  24.47  72.96 27.703
OBS: 60242.2370 i    WFD        0   296.80  2727.99     46 0.70 -1.064e+01  7.912e+00 31.849  33.02  72.49 27.693
OBS: 60246.1118 Y    WFD        0  3300.04  1663.68    119 0.70 -3.180e+01  3.943e+01 30.635  70.09  83.41 25.800
OBS: 60246.1154 Y    WFD        0  1800.46  3156.31    152 0.70 -4.419e+00  3.978e+01 30.634  69.61  83.99 25.799
OBS: 60247.3142 z    WFD        0  3806.44   284.98     38 0.70 -2.254e+01  2.535e+01 31.438 109.22  91.75 25.517
OBS: 60261.0729 u    WFD        0  3771.15  3769.02    119 0.70  3.062e+00  7.495e+00 30.464  87.32  11.20 29.449
OBS: 60261.0930 u    WFD        0  2754.77  1651.88    177 0.70 -4.901e-01  7.126e+00 30.463  81.93  11.12 29.449
OBS: 60269.1027 i    WFD        1   200.84  1966.69    179 0.70  1.800e+01  7.210e+00 31.848  44.71  55.70 24.186
OBS: 60273.0728 z    WFD        0  4008.60  2566.95     56 0.70  1.430e+01  1.774e+01 31.460  55.39  89.60 24.092
OBS: 60275.0443 z    WFD        0  1457.96  3192.94    101 0.70 -2.477e+00  2.608e+01 31.444  90.16 104.01 24.110
OBS: 60275.0452 z    WFD        0  3681.74  3178.59    124 0.70  2.975e+01  2.567e+01 31.444  87.91 103.66 24.110
OBS: 60278.2548 i    WFD        0  2391.14  3824.27    179 0.70  1.216e+01  9.350e+00 31.850  38.68  78.72 24.342
OBS: 60279.0504 g    WFD        0   182.70    53.90     38 0.70  2.433e+00  3.251e+00 32.315  94.25  26.41 28.454
OBS: 60279.0634 g    WFD        0   320.90   630.34     54 0.70  2.710e+00  3.117e+00 32.321  88.33  26.00 28.453
OBS: 60309.2123 i    WFD        0  1822.46  3293.39    183 0.70  1.460e+01  9.814e+00 31.847  51.02  73.48 25.969
OBS: 60309.2245 i    WFD        0   550.85  3609.07     49 0.70 -5.634e+00  9.843e+00 31.847  50.44  74.03 25.969
OBS: 60309.2276 i    WFD        0  3096.01   161.90    111 0.70 -2.052e+00  9.647e+00 31.849  47.91  74.42 25.970
OBS: 60328.0604 Y    WFD        0  3511.57   311.50     24 0.70  3.895e+01  4.837e+01 30.581 174.82  63.44 25.833
OBS: 60328.0622 Y    WFD        0    56.19  2797.22     12 0.70  4.358e+01  4.881e+01 30.579 178.21  63.29 25.833
OBS: 60330.1113 i    WFD        0  2822.18  2865.28    183 0.70 -7.031e-01  1.201e+01 31.835  72.94  75.22 27.032
OBS: 60336.0521 z    WFD        0  2050.38  3928.29    168 0.70 -6.559e+00  2.390e+01 31.420 143.07  74.48 26.524
OBS: 60336.0541 z    WFD        0  3851.30  2662.95    126 0.70 -2.223e+01  2.428e+01 31.413 146.57  74.59 26.524
OBS: 60338.1232 i    WFD        0  3571.42   804.06     19 0.70 -1.509e+01  1.195e+01 31.838  67.62  77.98 27.378
OBS: 60369.0307 z    WFD        0   927.90  3462.19      5 0.70  1.222e+01  1.363e+01 31.457  62.44  63.46 27.049
END:
```

How were data files created for SDSS and DES ?

- extract forced photometry + meta data from DB; write intermediate **TEXT** format … OR
- Final photometry writes **TEXT** formatted files
- SNANA utility converts **TEXT → FITS**
- TEXT files tarred up (or removed)
- What about LSST ?

Pros and Cons of Intermediate **TEXT** files

- Pros:
  - **TEXT** → **FITS** translator already exists
  - translator includes t_0 estimator that is robust to crazy fluxes
  - easier to debug since TEXT is always there
- Cons:
  - with multi-cores, might bump into file-count limits (e.g., 50k per FITS file x 10 cores → max 500k files)
  - slower (hunch is negligible compared to DB extraction)

Who created data files for previous Wide-Area Surveys ?

- Rick (SDSS,DES), Dan+David (PS1), Bruno(DC2), … ?
- Lots of survey team help loading DB, but little interest in planning/creating data files.
- Will Rick,Dan,David,Bruno perform this task for LSST ? Answer: ***NO*** (but we can help create a process)
- To my knowledge, this task has not been addressed in funding proposals, analysis planning, conferences.
- Should re-think ``data file'' status for LSST

Can we re-use previous makeDataFile codes ?

Inventory of accessible **makeDataFile** codes:

- SDSS
- DES
- DC2

Can we recycle previous makeDataFile codes ?

Inventory of accessible **makeDataFile** codes:

- SDSS
- DES
- DC2

Previous codes useful for experience, but not for LSST because :

- too tangled in their respective DB system
- not designed for multiple projects
- less work to re-write than to refactor

Strategic Suggestion for Data File Creation in LSST

- SNWG input on unresolved issues (next slides)
- Pipe Scientists deliver some utilities
- DESC-SN members take shifts (few months) during/after operations to
  - 1) maintain and update data files
  - 2) perform monitor tasks to ensure data quality is suitable for cosmology analysis.
  - 3) report problems to DM and Pipe-Sci's
  - 4) contribute to their builder status?

Unresolved Issues from Previous Surveys

- Implement ***never-ending small fixes***, e.g.,
  - fix handful of redshifts
  - fix crazy ZP in 3 patches
  - fix galaxy catalog bug for hosts
  - fix makeDataFile code bug
- Override data tables? e.g., fixed a few redshifts, try different hostMass and vPec (see OVERRIDE feature in SNANA manual)
- post-photometry corrections? e.g., chromatic, fluxErr scales; override fluxes with separate maps, or update data files ?
- Archive prelim data files used in publications (e.g., method papers)
- Manage desire for stability vs. desire to fix issues and add data.
- Organize WFD (new issue compared to SDSS,PS1,DES)
- Select LSST subset for photometric analysis (NOT Galactic, SN-like …)
- Answers may depend on time to remake data files (hrs, days, week?)

Override Features

- Pros:
  - greatly reduce frequency of remaking data files
  - multiple override options for one set of data files (e.g., test different codes for zSPEC & zPHOT)
  - easy to test ``not-so-standard'' items (e.g., VPEC, hostMass)

- Cons:
  - all analysis codes must replicate overrides
  - validation among codes ... how often ?
  - requires more organization

Override Features in SNANA

5.27.1 Over-Riding Header Information

Here we describe how to over-ride header information in the data files without re-making the data files. This feature can be useful, for example, for systematics or optimization studies The header information includes host properties, peculiar velocities, redshifts, and Galactic extintion, all of which can be easily modified for a particular study. To see a complete list of options, search for `VARLIST_ALL_HEADER_OVERRIDE` in `snana.car`.

The input key for the analysis programs is

```
&SNLCINP
  HEADER_OVERRIDE_FILE = 'myUpdates.dat'
      or
  HEADER_OVERRIDE_FILE = 'myUpdates1.dat,myUpdates2.dat
```

```
more myUpdates.dat
VARNAMES: CID  REDSHIFT_HELIO  REDSHIFT_HELIO_ERR MWEBV
SN:  1346137      0.246  0.001    0.021
SN:  1346387      0.533  0.0011   0.038
etc ...
```

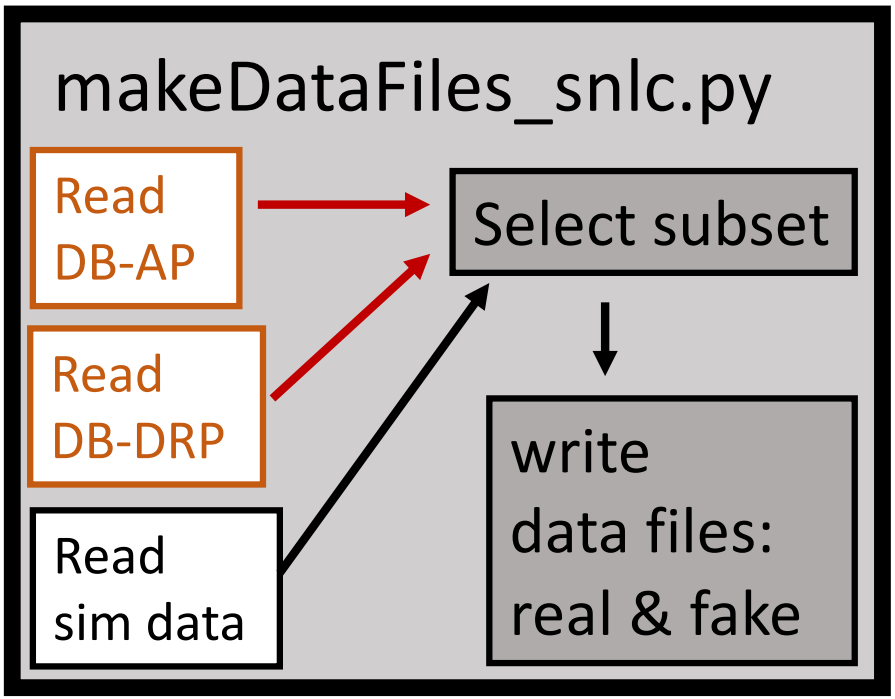
In the first example, three header values are updated: helio redshift, the uncertainty, and Galactic extintion. For SNe missing from the update file, the original values in the data file header are used. This feature does `NOT` work on epoch-dependent quantities. The second example illustrates a comma-separated list of table files; beware that the variables in each file must be unique (except for CID).

The header-override list includes variables that are used later in BBC or cosmology fitting, but not used in light curve fitting; e.g., host-galaxy properties and peculiar velocity. To avoid losing time re-running identical light curve fits for such variable changes, it can be more efficient to make substitutions at the BBC/`SALT2mu.exe` stage using input

```
datafile_override=over1.dat,over2.dat,etc
```

Any variable in the input table (see `datafile` arg) can be included in an override table. For `VPEC` (`VPEC_ERR`) changes, the Hubble diagram redshift (`zHD`, `zHDERR`) is re-computed by subtracting the original v_{pec} and adding the override value.

Suggested Architecture for DESC-SN + Pipe-Sci's



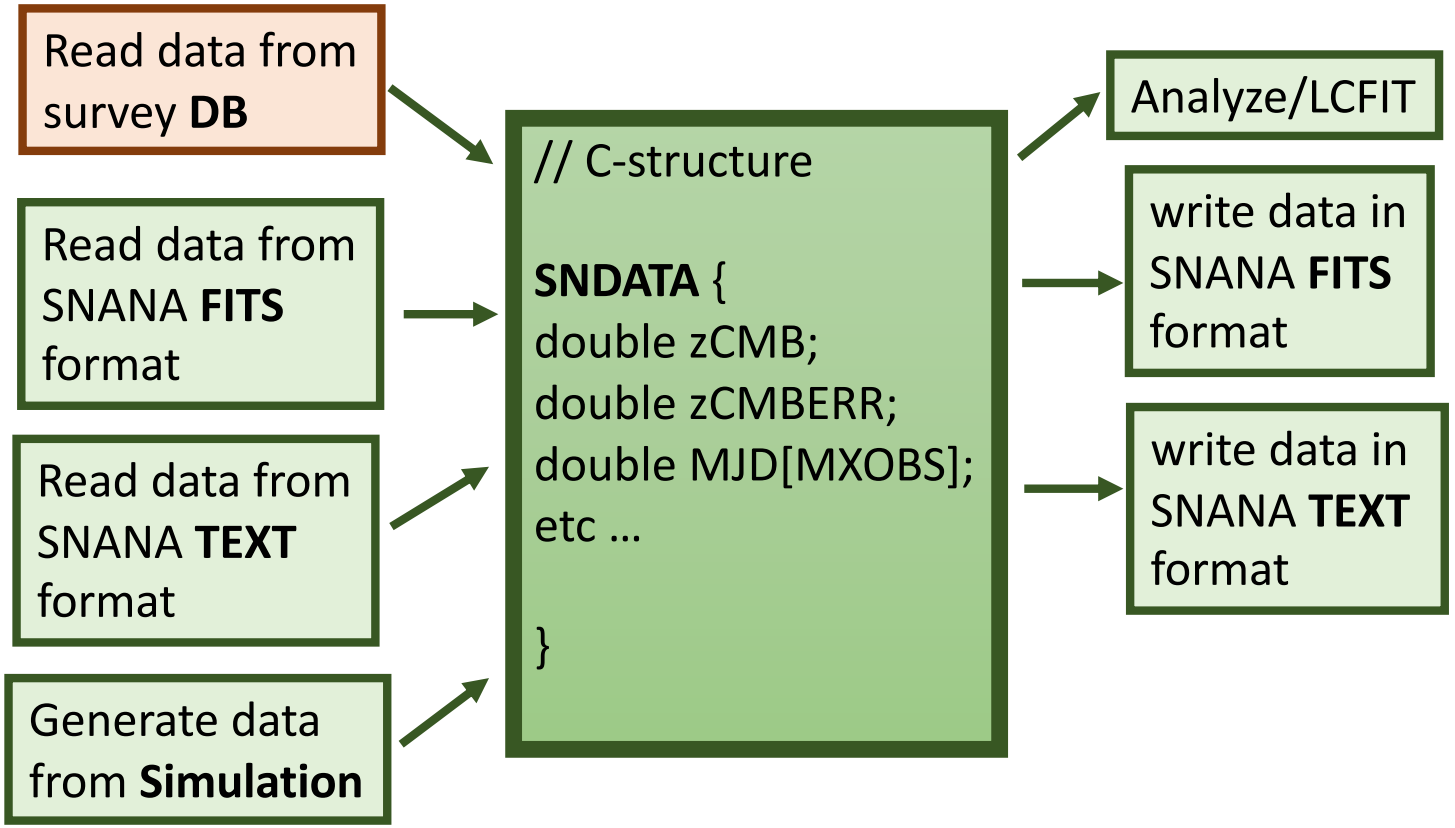
Monitor:

1) Before release, compare against previous data files and
   * quantify changes
   * catch disasters
2) compare data/sim distributions
3) update/init temporary overrides
4) archive previous version(s)

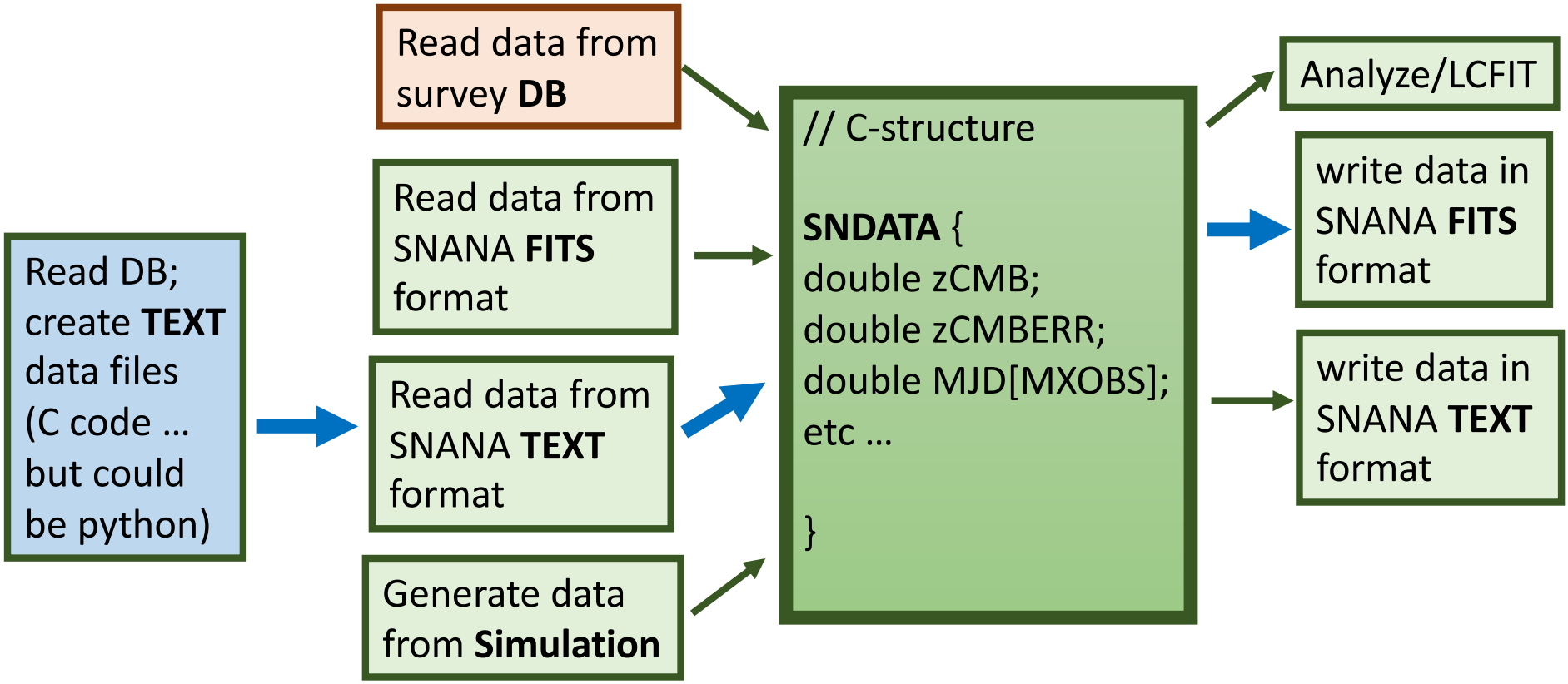
Key point:

development can & should begin long before DB exists

Appendix: SNANA C-code architecture



Appendix: DES-makeDataFiles



Appendix: LSST-makeDataFiles ?

?

?

Read DB; create **TEXT** data files (C or python)

Read data from survey **DB**

Read data from SNANA **FITS** format

Read data from SNANA **TEXT** format

Generate data from **Simulation**

// C-structure

SNDATA {
double zCMB;
double zCMBERR;
double MJD[MXOBS];
etc …

}

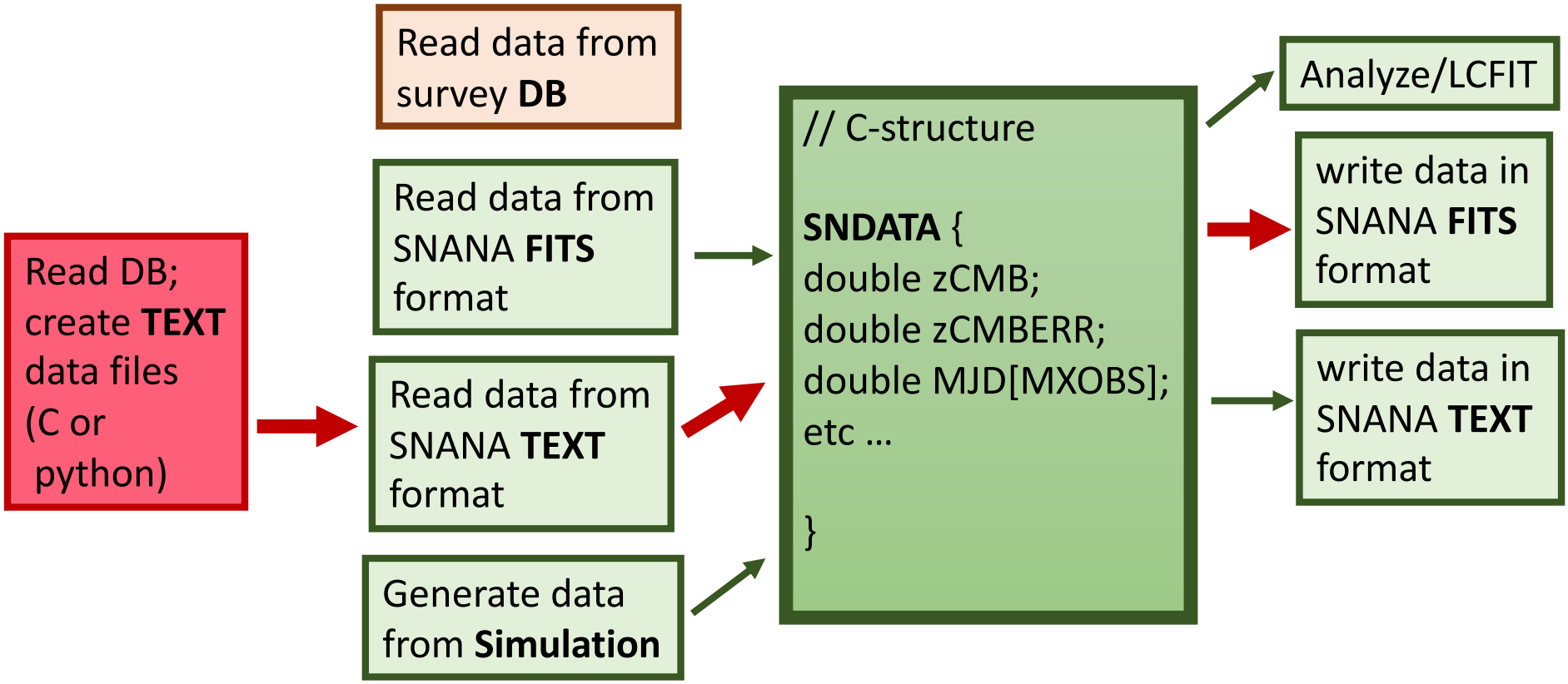
Analyze/LCFIT

write data in SNANA **FITS** format

write data in SNANA **TEXT** format

Elegance vs. *Practical*

Appendix: LSST-makeDataFiles ?



Elegance vs. *Practical*

Recommendation: Practical