

USINE propagation code and associated tools

1. Galactic cosmic rays
2. Transport equation and codes
3. USINE inputs
4. Structure and documentation
5. USINE output examples
6. Conclusions



DMAstro-LHC
ANR-12-BS05-0006



David Maurin
(LPSC)

dmaurin@lpsc.in2p3.fr



ICRC 2015
4/8/2015

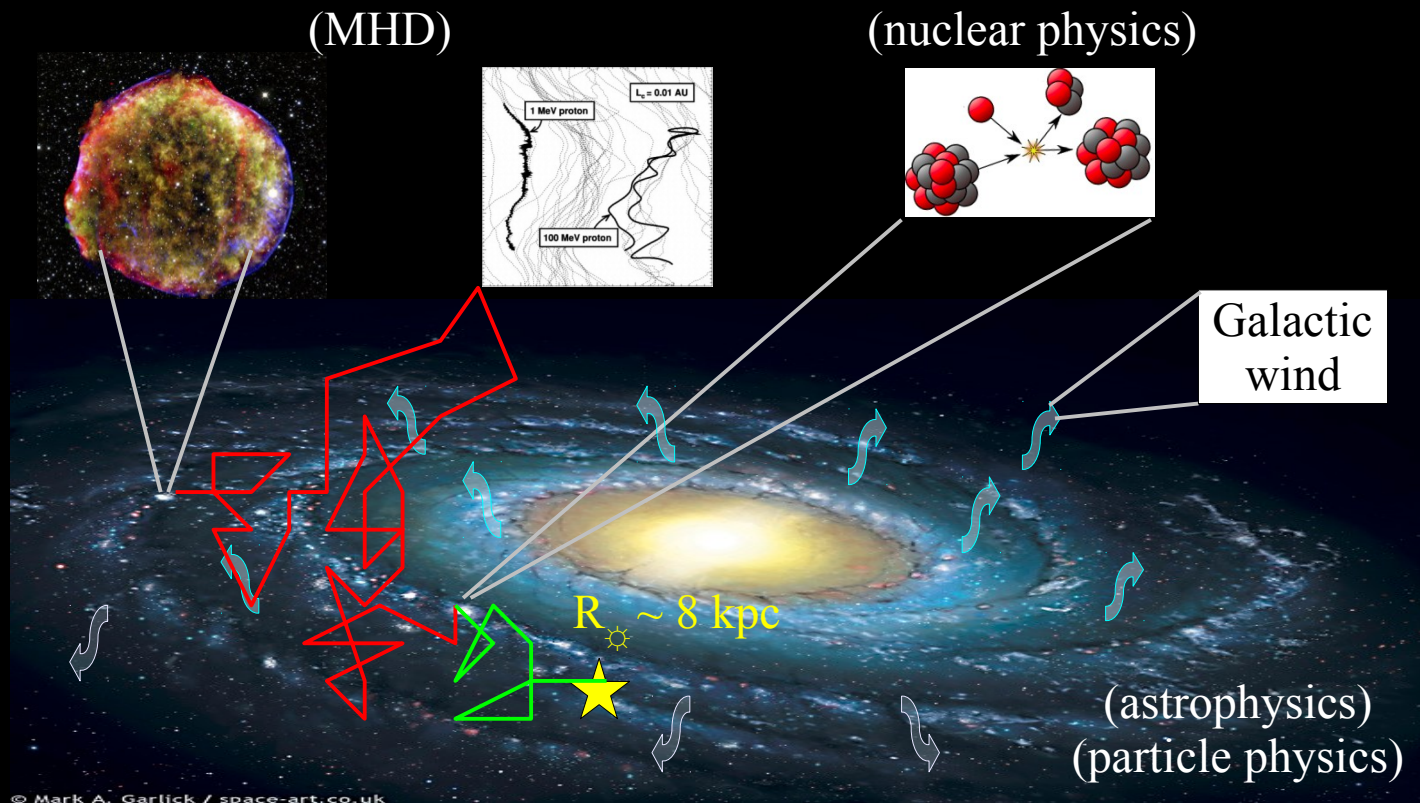
1. Galactic cosmic ray transport

1. Sources

- Spectrum
- Abundances
- Spatial distribution

2. Transport in the Galaxy

- diffusion
- convection
- energy gains/losses
- fragmentation/decay



→ **Goal: understand fluxes, search for dark matter**
N.B.: phenomenological approach (diffusion equation)

2. Transport equation and codes

$$\overbrace{\frac{\partial N^j}{\partial t}}^{\text{Variation}} + \overbrace{\left(-\vec{\nabla} \cdot \left(D(E, \vec{r}) \vec{\nabla} \right) + \vec{\nabla} \cdot \vec{V}_c(\vec{r}) \right)}^{\text{Spatial transport: diffusion+convection}} N^j + \overbrace{\frac{\partial}{\partial E} \left(b^j N^j - D_{EE} \frac{\partial N^j}{\partial E} \right)}^{\text{E losses and gains}} + \overbrace{\left(\Gamma_{\text{rad}} + \Gamma_{\text{inel}} \right)}^{\text{Catastrophic losses}} N^j = \overbrace{Q^j(t, E, \vec{r})}^{\text{Source}}$$

2. Transport equation and codes

$$\overbrace{\frac{\partial N^j}{\partial t}}^{\text{Variation}} + \overbrace{\left(-\vec{\nabla} \cdot \left(D(E, \vec{r}) \vec{\nabla} \right) + \vec{\nabla} \cdot \vec{V}_c(\vec{r}) \right)}^{\text{Spatial transport: diffusion+convection}} N^j + \overbrace{\frac{\partial}{\partial E} \left(b^j N^j - D_{EE} \frac{\partial N^j}{\partial E} \right)}^{\text{E losses and gains}} + \overbrace{\left(\Gamma_{\text{rad}} + \Gamma_{\text{inel}} \right)}^{\text{Catastrophic losses}} N^j = \overbrace{Q^j(t, E, \vec{r})}^{\text{Source}}$$

| | <i>(Semi-)analytical</i> | <i>Numerical</i> | <i>Monte Carlo</i> |
|--------------------------------|--|--|---|
| Approach | <u>Simplify the problem:</u> <ul style="list-style-type: none"> • keep dominant effects only • simplify the geometry | <u>Finite difference scheme:</u> <ul style="list-style-type: none"> • discretise the equation • scheme (e.g., Crank-Nicholson) | <u>Follow each particle:</u> <ul style="list-style-type: none"> • N particles at t=0 • evolve each of them to t+1 <p>1D : $\Delta z = \pm \sqrt{2D\Delta t}$</p> |
| Tools | <ul style="list-style-type: none"> • Differential equations (Green functions, Fourier+Bessel expansions...) | <ul style="list-style-type: none"> • Numerical recipes/solvers (NAG, GSL libraries) | <ul style="list-style-type: none"> • Stochastic differential equations (Markov process) + MPI |
| Pros | <ul style="list-style-type: none"> • Useful to understand the physics • Fast (MCMC analyses “simple”) | <ul style="list-style-type: none"> • Very simple algebra • Any new input easily included | <ul style="list-style-type: none"> • Statistical properties (along path) • No grid but t step (for/back)-ward |
| cons | <ul style="list-style-type: none"> • Only solve approximate model • New solution for new problem | <ul style="list-style-type: none"> • Slower, memory for high res. • “Less” insight in the physics | <ul style="list-style-type: none"> • Even slower (+ statistical errors) • Massively parallel problem |
| Codes and/or references | Webber (1970+) Ptuskin (1980+) Schlickeiser (1990+) USINE (2000+) | GALPROP (Strong et al. 1998) DRAGON (Evoli et al. 2008) PICARD (Kissmann et al., 2013) | Webber & Rockstroh (1997) Farahat et al. (2008) Kopp, Büshing et al. (2012) |

3. USINE inputs

Welcome Experiments/Data Data extraction Links New data <http://lpsc.in2p3.fr/crdb>

Database of Charged Cosmic Rays

D. Maurin (LPSC), F. Melot (LPSC), R. Taillet (LAPTh)
If you use this database, please cite [Maurin, Melot, Taillet, A&A 569, A32 \(2014\)](#)

New release V2.2 - December 2014
[\[changelog\]](#)
Last code modification: 10/06/2015

Accessing the database

- Experiments/Data: list of experiments, publications, data
- Data extraction: selection by flux/ratio/energy range... (on this web site or via a REST interface)
- Export database content in **USINE** or **GALPROP** compliant format (ASCII files)
- Get all bibtex entries and Latex cite (by sub-experiment)

1. CR data

USINE formatted ASCII file(s) from CRDB

→ Easy to add his own data (for a propagation analysis)

2. Production cross-sections

USINE formatted ASCII files: inelastic, straight-ahead, differential production, non-annihilating...)

→ accommodate for any ISM # of targets in calculation (as long as X-section provided)

```
# Type Unit Targets
sigma mb H, He
# EknMIN[GeV/n] EknMAX[GeV/n] nEkn (PROJECTILE GRID)
5.000000e-02 6.294626 43
70Zn -> 68Zn
5.483168e+01 8.574767e+01
```

3. Propagation set-up + model parameters

- CR list/charts, energy grids, targets...
- CR data and cross-section files
- Geometry, ISM description
- Source+transport (formulae, splines...)
- Free pars (collected for minimisation)**

→ ASCII initialisation files with web interface
http://lpsc.in2p3.fr/usine_gui (beta version)

Transport - Hide

Info: Description of the transport is always (i) free parameters (+ units and vals); (ii) space/time-dependent values/formule (up to 3 components); (iii) space/time-dependent scalar for Alfvén speed and momentum diffusion (up to 9 components) for spatial diffusion

| Name | Type | Values | Unit |
|----------|--------|--|-------------|
| ParNames | string | Va,Vc,K0,delta,eta_t | [-] |
| ParUnits | string | [km/s],[km/s],[kpc^2/Myr],[],[] | [-] |
| ParVals | string | 70.,15.,0.0112,0.7,1. | [-] |
| Wind* | double | W0:1 Vc | [km/s] |
| VA | double | 1 Va | [km/s] |
| K* | string | K00:1 beta^eta_t*K0*Rig^delta | [kpc^2/Myr] |
| Kpp | string | 1 (4./3.)*(Va*1.022712e-3*beta*Etot)^2/(delta*(4-delta)^2*(4-delta)*K00) | [GeV^2/Myr] |

→ Lots of parameters for a run: ASCII files and graphic interfaces help sort them out

4. Structure and documentation

USINE V.3 (C++, Root CERN libraries)

*Read CRDB
exported files*

CR and nuclear inputs

TUAtomElements.h
TUAtomProperties.h
TUCREntry.h
TUCRList.h
TUXSections.h

ISM

TUMediumEntry.h
TUMediumTXYZ.h

Transport

TUTransport.h

CR sources

TUSrcVirtual.h
TUSrcPointLike.h
TUSrcSteadyState.h
TUSrcMultiCRs.h
TUSrcTemplates.h

CR data

TUDataEntry.h
TUDataSet.h
TUDatetime.h
TUNormEntry.h
TUNormList.h

Init. + free params

TUInitParEntry.h
TUInitParList.h
TUFreeParEntry.h
TUFreeParList.h

- Finished (tested+documented)
- To be documented

Axes/handlers

TUAxis.h
TUAxesCrE.h TUCoordE.h
TUAxesTXYZ.h TUCoordsTXYZ.h
TUValsTXYZVirtual.h
TUValsTXYZFormula.h
TUValsTXYZGrid.h
TUValsTXYZECr.h

Propagation models

TUModelBase.h
TUModel0DLeakyBox.h
TUModel1DKisoVc.h
TUModel2DKisoVc.h { TUBesselJ0.h
 TUBesselQiSrc.h

Solar modulation models

TUSolModVirtual.h
TUSolMod0DForceField.h
TUSolMod1DSpherSym.h

TUPropagSwitches.h
TURunPropagation.h

./bin/usine_run

Utilities

TUGlobalEnum.h
TUMessages.h TUQueries.h
TUNumMethods.h
TUMath.h TUPhysics.h
TUPhysicsProcesses.h

→ Flexible toolbox to ease the analysis/development of propagation models

4. Structure and documentation

ROOT-style (THtml)

Doxygen-style

```
class TUModelBase: public TUAxesCrE,  
TUPropagSwitches, public TUDataset, p  
TUNormList, public TUSrcTemplates, pu  
TUXSections, public TUFreeParList
```

TUModelBase

Base ingredients for all propagation models (CR data, X-sections...).

The class `TUModelBase` is the centrepiece of all propagation models. It contains model-independent ingredients (inherited) and class members ingredients. The first category corresponds to quantities that are somehow independent of the propagation model selected (such as CR data, X-sections,...). The second category encompasses quantities that define the propagation model (geometry, transport, sources...). We detail below (I. and II.) the content of these two categories. We present separately (III.) the classes `TUFreeParList` (inherited) and `TUSolModVirtual` (class member), because of their specific role in this class (and for propagation models in general). To conclude, we provide a brief 'How to' (IV.) on how to write a class for a propagation model deriving from this class.

Data Members

private:

```
TUAxesTXYZ* fAxesTXYZ  
TUCoordE* fCoordE  
TUCoordsTXYZ* fCoordsTXYZ
```

Model geometry
Generic E coordinate (bir
Generic TXYZ coordinate

The screenshot displays a web-based documentation interface in a Doxygen style. At the top, there are tabs for 'Main Page', 'Namespaces', 'Classes', and 'Files'. The 'Classes' tab is active, showing a 'Class List' with a brief description: 'Here are the classes with brief description'. A search bar at the top right contains the text 'GetNa'. A dropdown menu is open, showing a list of search results including 'GetNamePar', 'GetNameROOT TUCRLList', 'GetNamesPureSecondaries TUCRLList', 'GetNameSubGroup TUInitParList', 'GetNameSwitch TUPropagSwitches', 'GetNameSwitches TUPropagSwitches', 'GetNameTarget TUXSections', 'GetNameTargets TUXSections', 'GetNameTertiaries TUXSections', 'GetNameTertiary TUXSections', 'GetNameTEXT TUCRLList', and 'GetNAxes TUAxesTXYZ'. Below the search results, a list of classes is shown, each with a circular icon and a name: 'TUAAtomEle', 'TUAAtomProperties', 'TUAxesCrE', 'TUAxesTXYZ', 'TUAxis', 'TUBesselJ0', 'TUBesselQiSrc', 'TUCoordE', 'TUCoordsTXYZ', 'TUCREntry', 'TUCRLList', 'TUDataEntry', 'TUDataset', 'TUDatetime', 'TUFreeParEntry', 'TUFreeParList', 'TUInitParEntry', and 'TUInitParList'.

→ Full documentation of classes/methods

5. USINE output examples

```
./bin/usine_run Usine.InitFiles/Usine.InitFiles/init.1DModel.par
```

```
-----  
INITIALISATION DONE  
=> READY FOR CALCULATIONS/DISPLAYS  
-----
```

```
+++++  
+ USINE Menu +  
+++++
```

A) PROPAGATION MODEL RESULTS (PRINTS & PLOTS)

A1. Local IS and TOA CR fluxes [A1+ to multiply flux by E^a , A1++ for extra plots]

A2. Local IS and TOA CR flux comparisons w/o decay/losses/etc. [A2+ to normalise to data for each config.]

A3. CR fluxes spatial distribution (1D or 2D depending on models selected)

A4. Fractions of secondary productions (per species)

A5. BETA-decay species [A5+ to check decayed=appeared]

B) MODIFY MODEL PARAMETERS (B to modify all)

B1. Propagation switches

B2. Transport parameters

B3. CR source parameters

C) INFO ON MODEL AND INPUT PARAMETERS (C to print all)

C1. Models (propagation and modulation)

C2. Geometry

C3. Transport parameters

C4. Propagation switches

C5. CR sources

C6. ISM

C7. CR list and parents (and E grids)

C8. CR and normalisation data

C9. X-section files and targets

E) CHI2 ANALYSIS

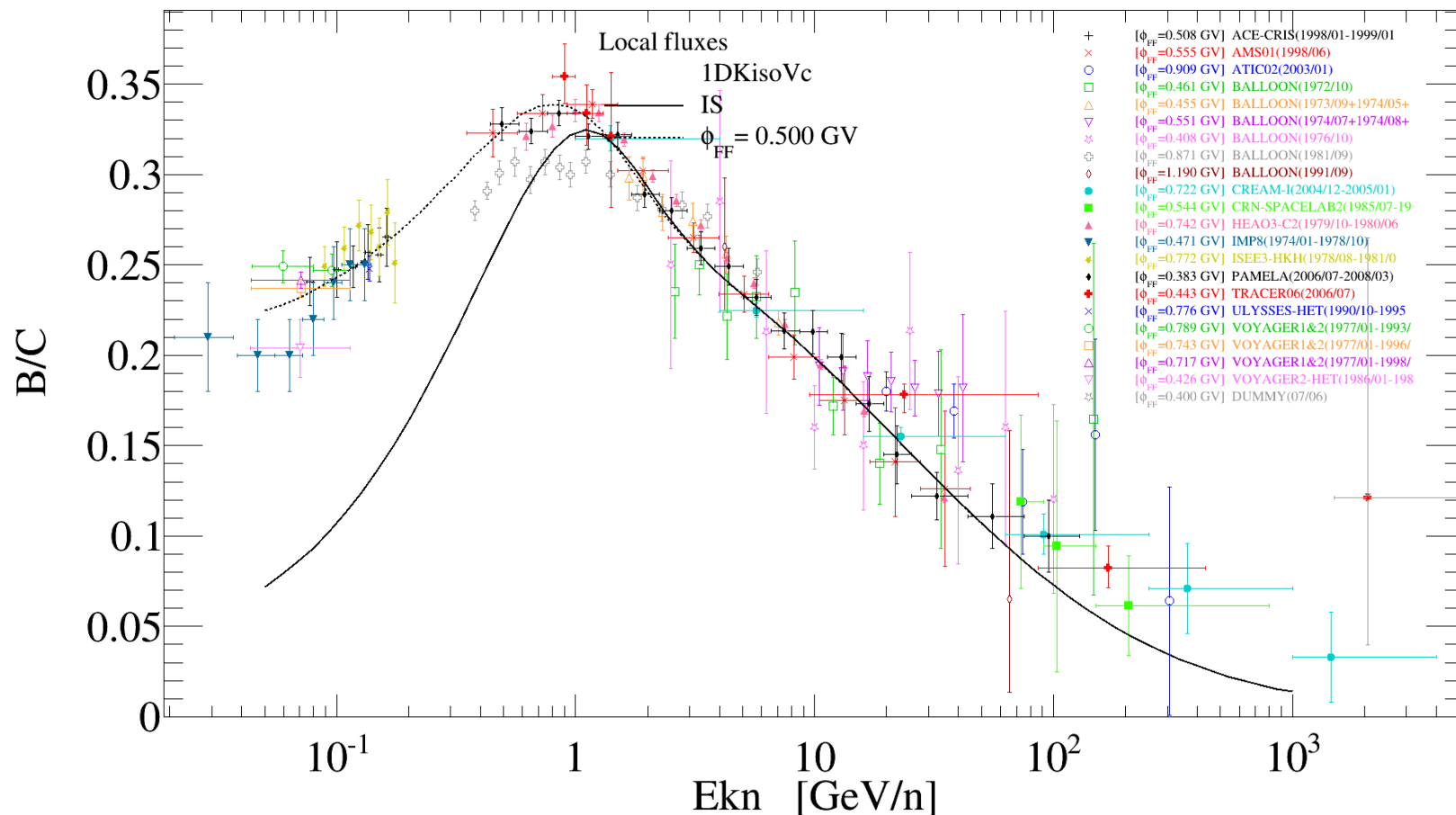
[Q to quit]

→ **Outputs = plots (+ ASCII and .root files [in progress])**

>> selection [e.g., A1]: █

5. USINE output examples: display IS and TOA

```
### Select a list of quantities (comma-separated, case insensitive)
    [0/Q= Quit]  ["List"= display list of available CRs and elements]
Valid quantities: 10B+11B,B/C,0 or <LNA> or ALLSPECTRUM
>> B/C
Element B is: 10B 11B
Element C is: 12C 13C 14C
### Select a comma-separated list (e.g., 0.,0.5,1) of modulation param phi (PHI=|Z|/A*phi) GV
>> .0,.5
### Fluxes will be multiplied by E^a
>> .0
```

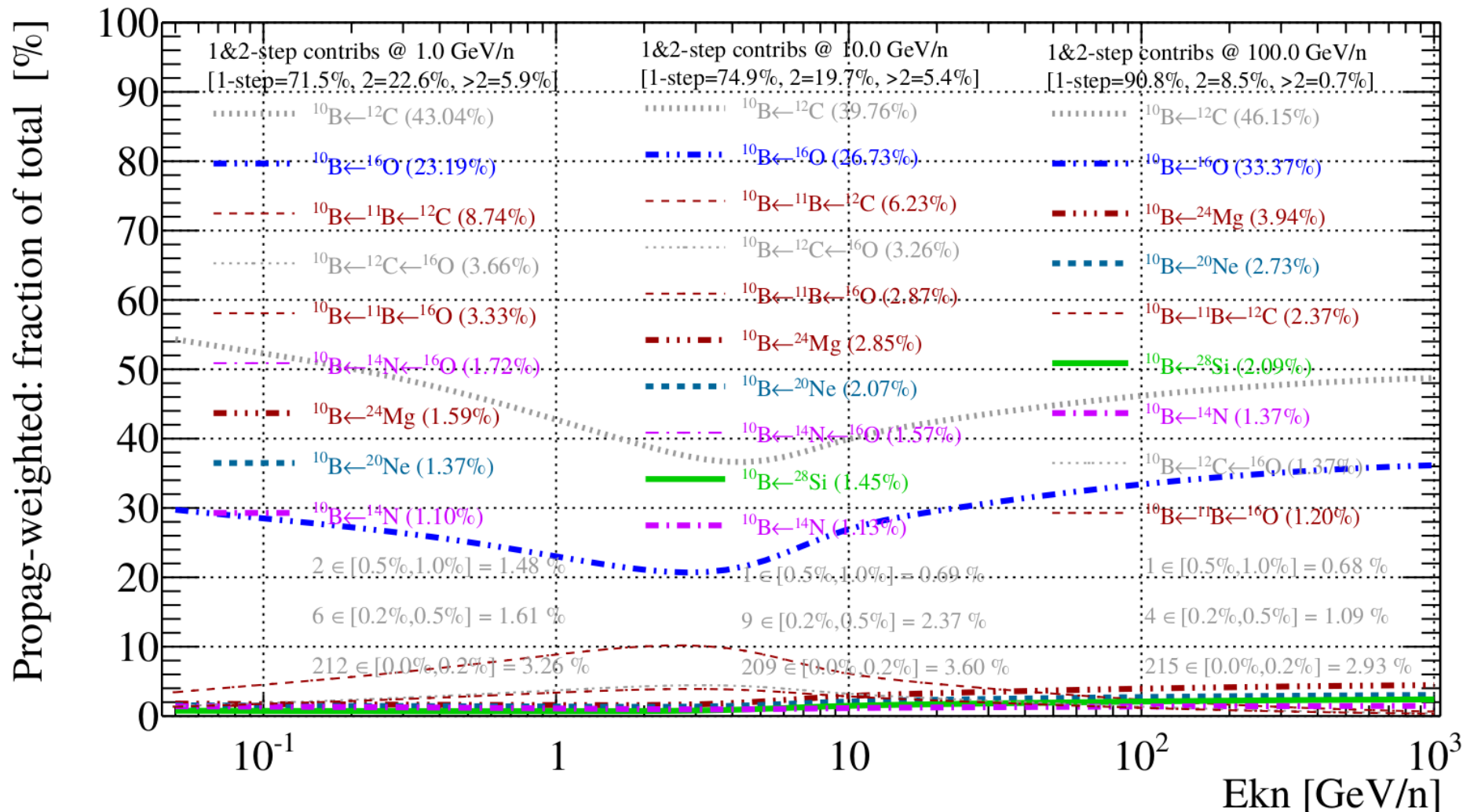


→ Quick display of any TOA combination (flux, ratio) isotopes and elements + data

5. USINE output examples: secondary production

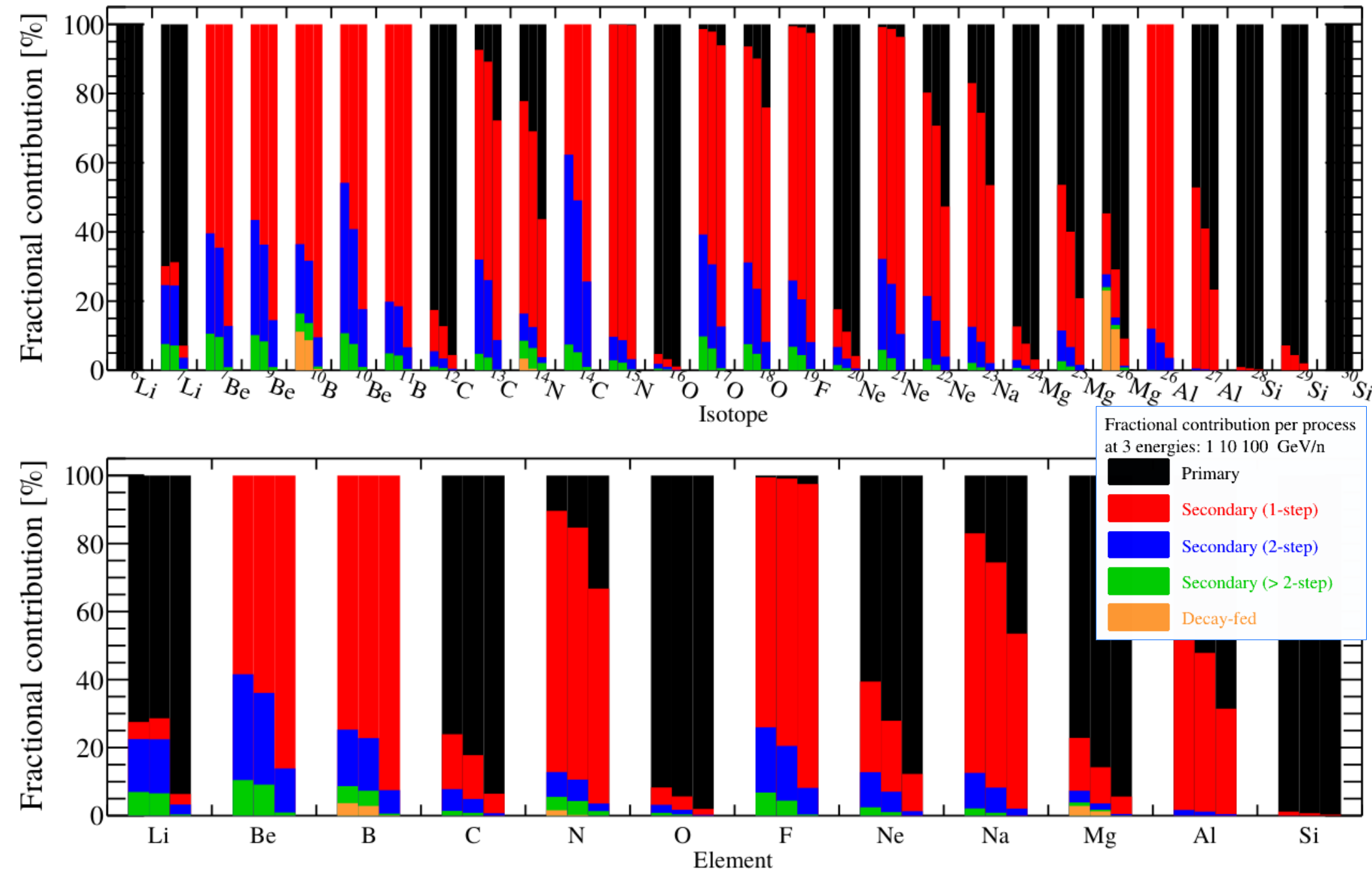
```
>>> Select Z (e.g., 6) or threshold (e.g., 0.8)
- all CRs with Z displayed
- only CRs whose secondary content is above threshold
  are considered (sec. content calculated @ Ekn~1 GeV/n)

>>> Select quantity (|integer|>1, or 0<threshold<1): 5
```



→ Reactions whose cross section matter most for the production of ^{10}B in GCRs

5. USINE output examples: relative contributions



→ Fractional contribution per process at three energies (for a source/transport model)

6. Conclusions

- **What does USINE do**
 - GCR propagation of nuclei and antinuclei
 - Solar modulation (Force-Field)
 - General toolbox (manage inputs and outputs)
- **Why/when should you use USINE?**
 - Training in GCR physics
 - To test/study impact of new X-sections
 - To test new semi-analytical models
 - When speed matters (e.g., for MCMC analyses)

- **Available now** (beta version)
 - From antideuterons to Zn
 - Content: leaky-box, 1D model (thin disk+halo)

git clone <https://gitlab.in2p3.fr/david-maurin/USINE.git>

- **By the end of August**
 - 2D model [to allow for DM pbar and dbar]
 - 1D spherically symmetric modulation
 - χ^2 minimisation, interface with MCMC engine
- **By the end of the year (if possible)**
 - Electrons and positrons (in 1D model)
 - Improved http://lpsc.in2p3.fr/usine_gui interface
 - $Z > 30$
 - ...

CRDB v3.1 (in prep.)

→ upper limits, antinuclei, $Z > 30$
+ HE?

Solar modulation (in prep.)

→ value for any time period
(based on neutron monitors)

+