

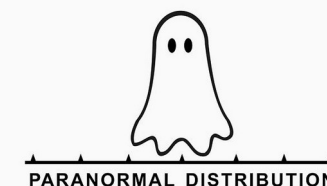
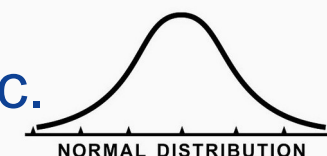
Data Handling & Analysis

(Groupe de Réduction et d'Analyse des Données Expérimentales de SOLEIL)

E. Farhi – GRADES/EXP/SOLEIL

We produced 3 PB in 2024. Currently limited by disk and network speeds.

- **Access:** Globus. 3 years embargo.
- **Catalog:** SciCat + DOI.
- **Nomenclature:** NeXus, NFFA/NFDI, Big-Map, PanOSC/OSCARS
 - ~ standardize metadata.
- **Data Format:** mostly NeXus.
 - ~ Data Type: 1D/vector, 2D/images, 3D/volume, hyper-spectral, etc.



Issue : our central storage is reaching its max ingestion rate.

We provide all inclusive Data Reduction, Analysis and Visualization services:

Thousands of software and libraries.

Debian packaging (+357) for **stability, science stack** and low maintenance.

We also support conda/pip and containers.

<https://salsa.debian.org/pan-team/soleil-packaging-overview>



DARTS (“low-tech” VISA – easy install/deploy) → **VISA**

<https://data-analysis.synchrotron-soleil.fr/desktop>

JupyterHub (shared storage with *DARTS*).

<https://data-analysis.synchrotron-soleil.fr/notebook/>

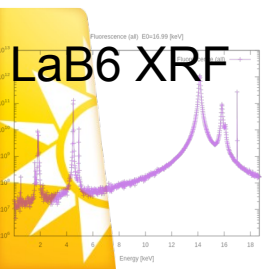
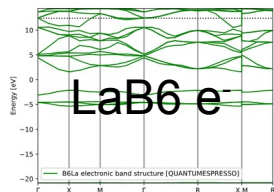
Material modeling (easy DFT+ e^-/ph dispersions)

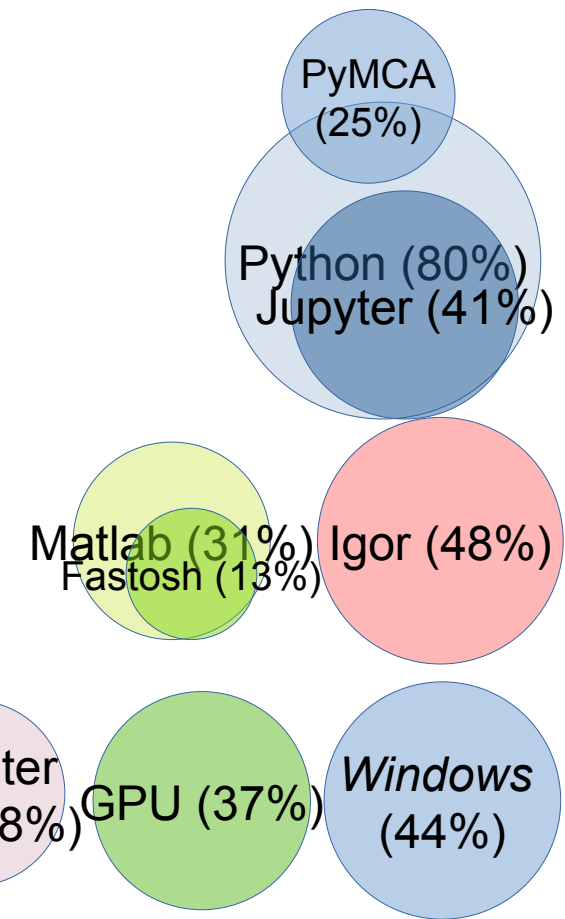
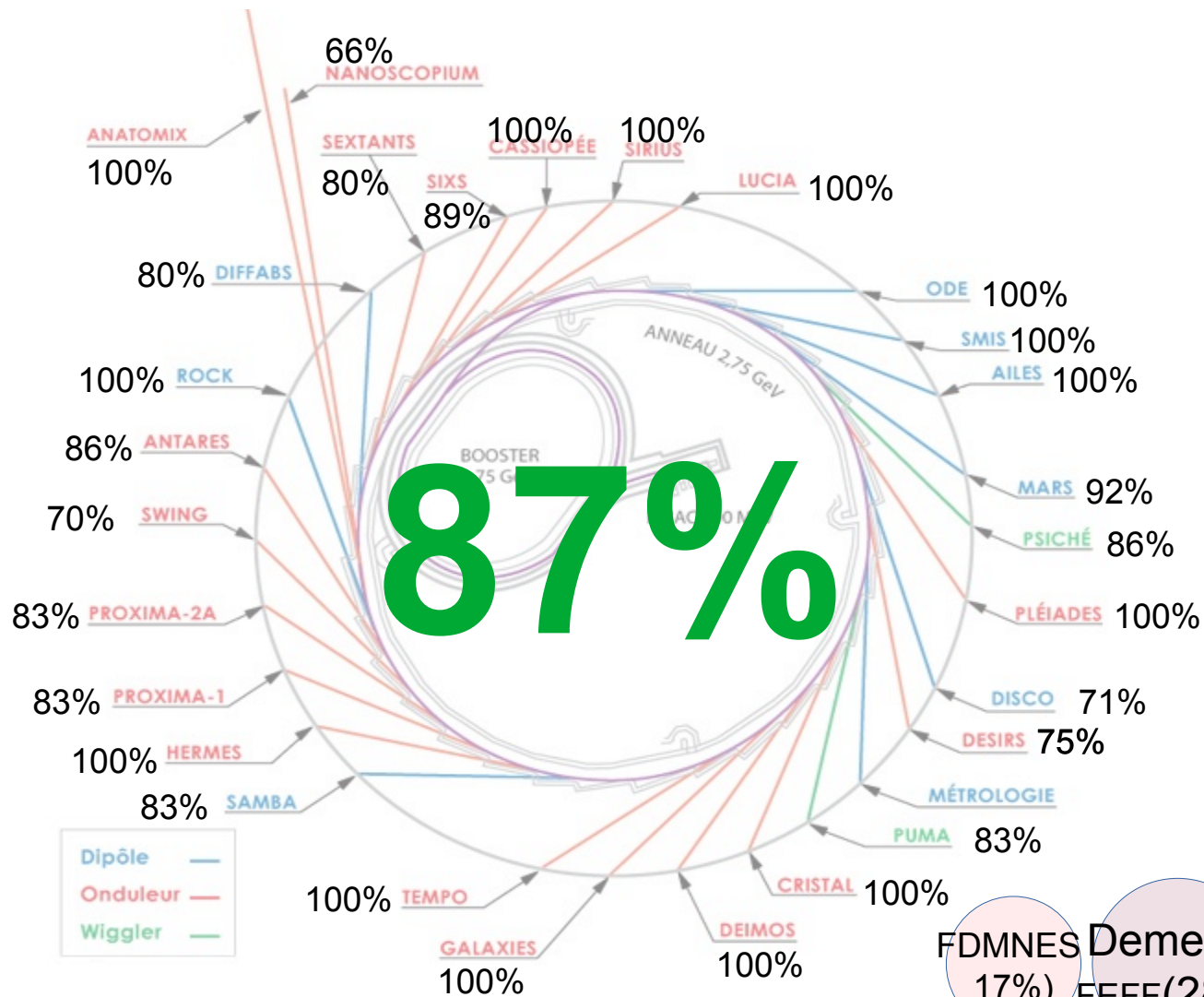
https://data-analysis.synchrotron-soleil.fr/computing/material_modeling.html

Diffraction + Fluorescence estimates

https://data-analysis.synchrotron-soleil.fr/computing/fluo_diffraction.html

LLM (local Chat – keep data safe. llama.cpp)





We maintain 20 scientific servers on BL
+ 3 central servers.

Be cautious with A.I.

- Requires massive annotated data.
- Must keep property of the data (no ChatGPT !)
- How to estimate the error bar on an A.I. result ?



We provide hardware and software:

- CPU + GPU, libraries (PyTorch/TF) in DARTS/JupyterHub.

Training data: BL+sample digital-twin:

- Fluorescence/XRF → Compound.
- Diffractogram/XRD → Space-group.
- SAXS → Large-structure model (SASView).
- XAS → Oxidation/coordination.

Will be provided as Edge-computing and cloud service.



Scientist's satisfaction is our major criteria.

- Balance: anarchy ("I want to do everything") over control ("No you can't do that").

Stability of software should be promoted. Our way → Debian

- Not there with conda/pip.

Maintenance should be minimized. Our way → Debian

- Duplicate work when using apptainer

Deployment should be eased. Our way → Debian and then Apptainer

- Users can install conda/pip on top

How do we manage ?

1) We contribute to scientific software.

2) We are DD/DM.

3) we use 40k€/y in Debian subcontracting. *Join us ?*

4) We search for simplicity – always. No complex IT stuff (we are **not** IT guys).