Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

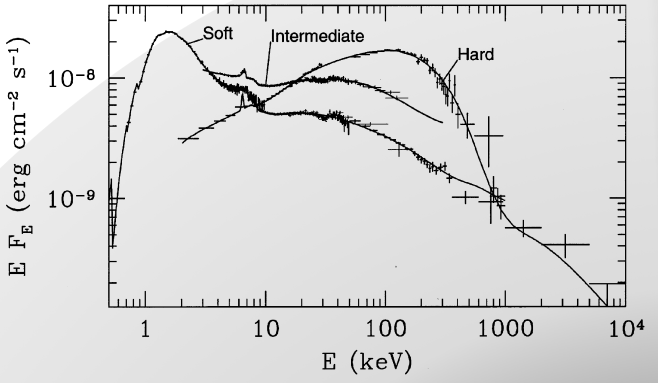
Stingray: Next-Generation Spectral Timing

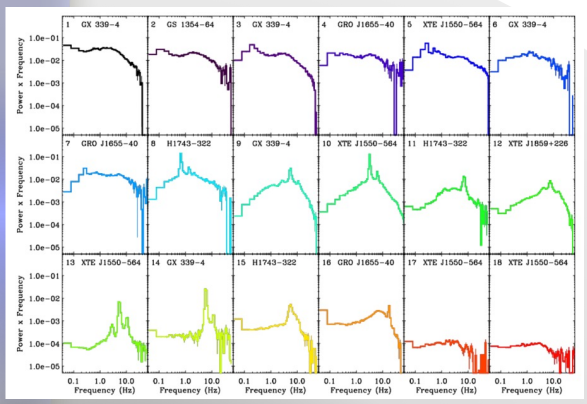
Eleonora Veronica Lai, Matteo Bachetti, Maura Pilia,
+ Daniela Huppenkothen and Stingray developers

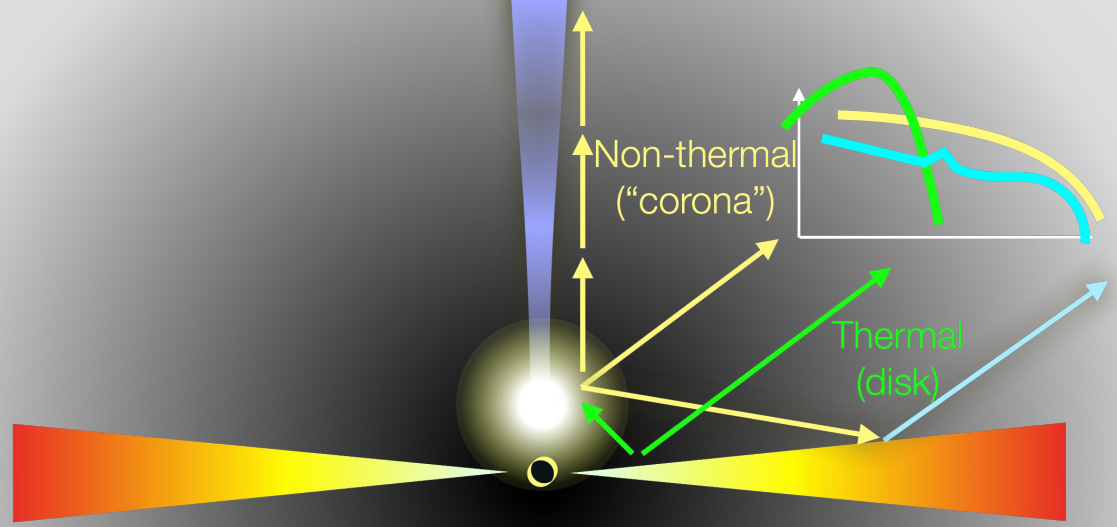
Spoke 3 Technical Workshop, Trieste October 9 / 11, 2023

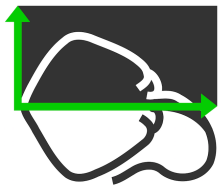
Scientific Rationale

- Some observe spectra, some observe variability. Is it possible to use the full information?
- Example: a variable accretion flow that **propagates** through an atmosphere (corona), that **illuminates** the accretion disk and gets **reflected**. Can we disentangle the emission regions?
- Stingray: ease the learning curve for advanced spectral-timing techniques, with a correct statistical framework









Huppenkothen et al. (2019)

History

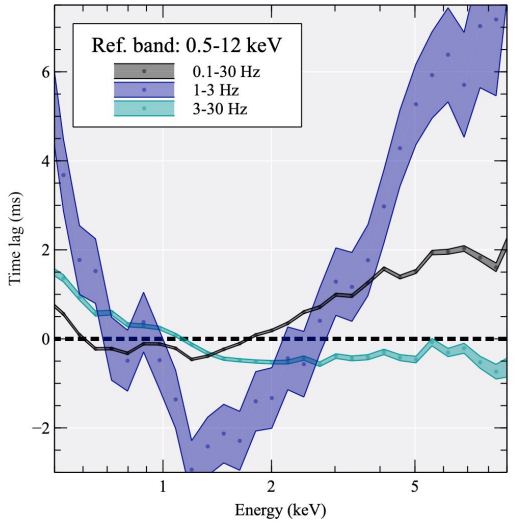
- **2016**, Leiden workshop „The X-ray Spectral-Timing Revolution": **Daniela Huppenkothen, Matteo Bachetti, Abigail Stevens, Simone Migliari, Paul Balm** decide the fusion of three existing packages for spectral timing, creating the **Stingray** library and two packages for interactive analysis based on it: **HENDRICS** (batch scripts) and **DAVE** (GUI)
- At the time, various official software packages for X-ray spectral fitting (e.g. XPSEC, ISIS, …), but **no open, well-maintained software for timing or spectral timing**

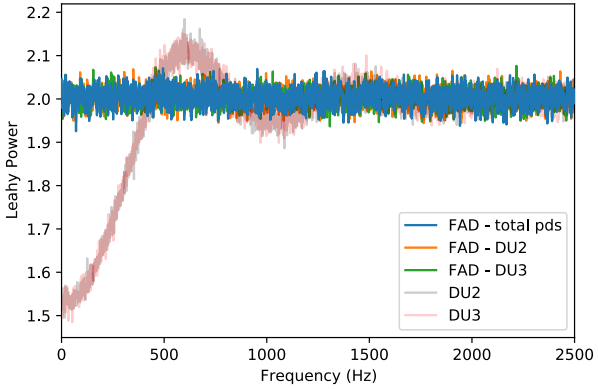
Spectral analysis	Timing analysis (+ lags)	Spectral timing
•Xspec •Sherpa •ISIS •(…)	•(XRONOS) POWSPEC •SITAR, Isisscripts.sl	X

- Today: Stingray is widely used in the X-ray community… even for things **that were not originally planned** (e.g. radio and optical data!)
- Significant code contributions from the community and Google Summer of Code students.

Technical Objectives, Methodologies, and Solutions: what we want

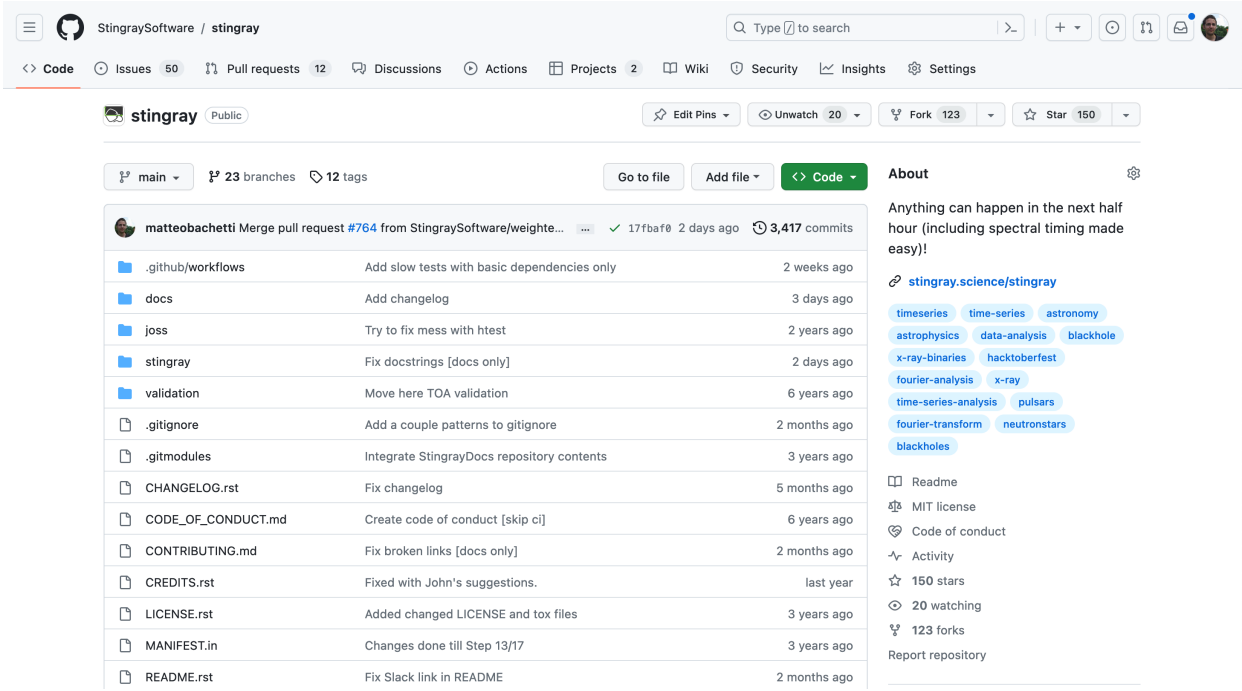
- **"Timing" analysis**
  - Pulsation searches and timing
  - Aperiodic variability, periodogram modelling (ML, Bayesian)
- **Spectral analysis -> connect to Xspec, Sherpa**
  - Continuum modeling
  - Broad lines (e.g. Fe complex, cyclotron lines)
- **Polarimetry (be creative!)**
- **All mixed together! E.g.**
  - Time lags
  - Spectral covariance, spectral polarimetry
  - Phase/Time-resolved spectroscopy and polarimetry
  - Time-resolved-energy-resolved polarimetry (Whatev')
- **... all with instrument awareness**
  - Be aware of instrumental systematics: dead time, frame time, good time intervals, etc.
  - Mission support





Technical Objectives, Methodologies, and Solutions: an open development model

- **Github-based** workflow:
  - Issue tracking
  - Assignments
  - Pull Requests
- Community outreach:
  - Public **Slack** channel
  - Talks
  - Hackatons/Tutorials
  - OpenAstronomy involvement
  - Astropy affiliated package
- Developers:
  - Astronomers
  - Google Summer of Code students



UNIAM
PRIN MIUR 2017

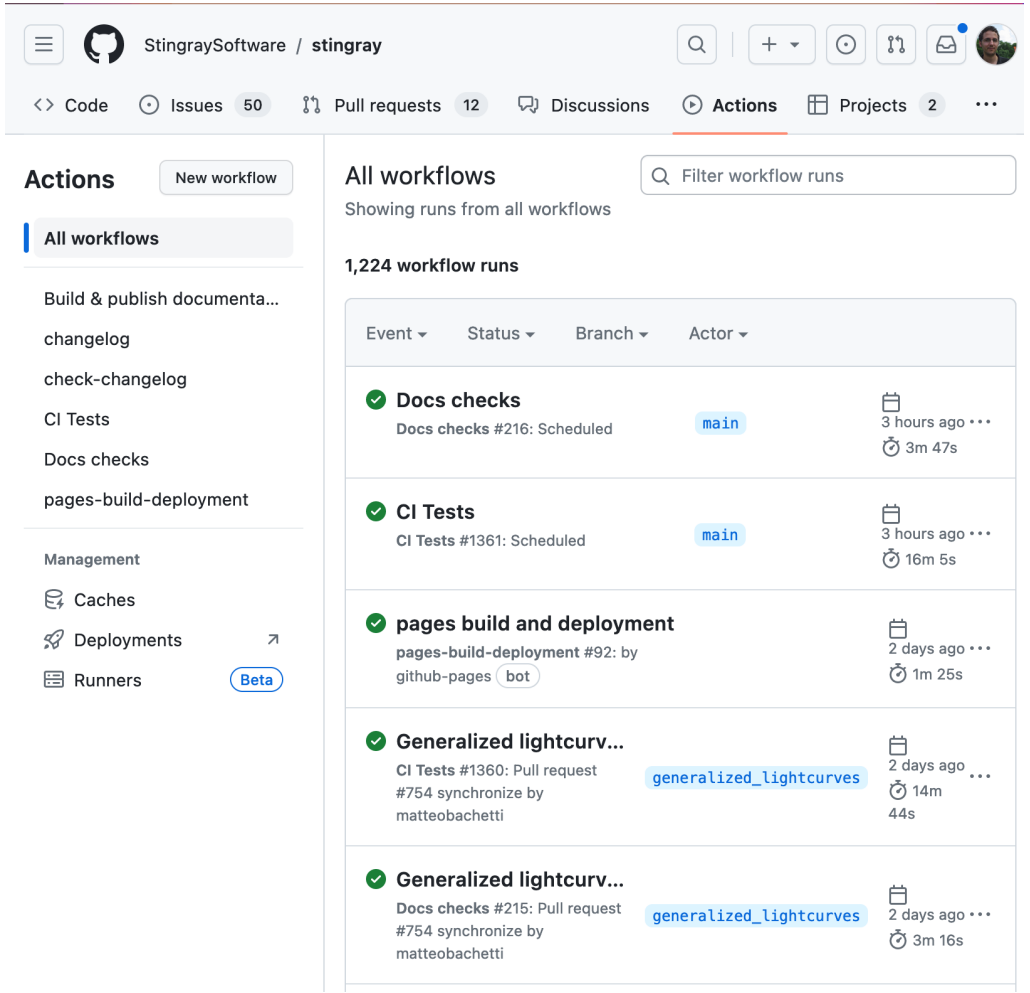
SpecTemPolar!
PRIN INAF 2019

INAF
ISTITUTO NAZIONALE DI ASTROFISICA

Google Summer of Code 2016, 2017, 2018, 2020, 2021, 2022, 2023

Technical Objectives, Methodologies, and Solutions: reliability and performance testing

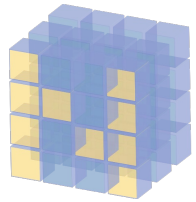
- Code correctness
  - Test-based development
  - Literature reproduction
- Regression testing: continuous integration with **Github Actions** and **tox**
  - **Unit tests**
  - **Integration tests**
- Performance
  - Profiling: **line_profiler**, **%time**, **memory_profiler**, etc.
  - Small-dataset testing (< RAM): verify "acceptable" execution times
  - Scalability for larger-than-RAM datasets
- Documentation
  - Use **Sphinx** + **Github Actions** for automatic docs building
  - **Linkcheck** for periodic link checking in the docs



Technical Objectives, Methodologies, and Solutions: performance improvement

- Pre-CN:
  - **Benchmarking**, **profiling**: single out bottlenecks
  - Code speedup technologies:
    - Algorithmic optimization
    - Use of optimized libraries when available
    - Just-in-time compilation (**JIT**) through **Numba**
    - Support of high-performance I/O libraries (**hdf5**, **netcdf**)

- CN (under construction):
  - Parallelization
  - Distributed computing
  - GPUs



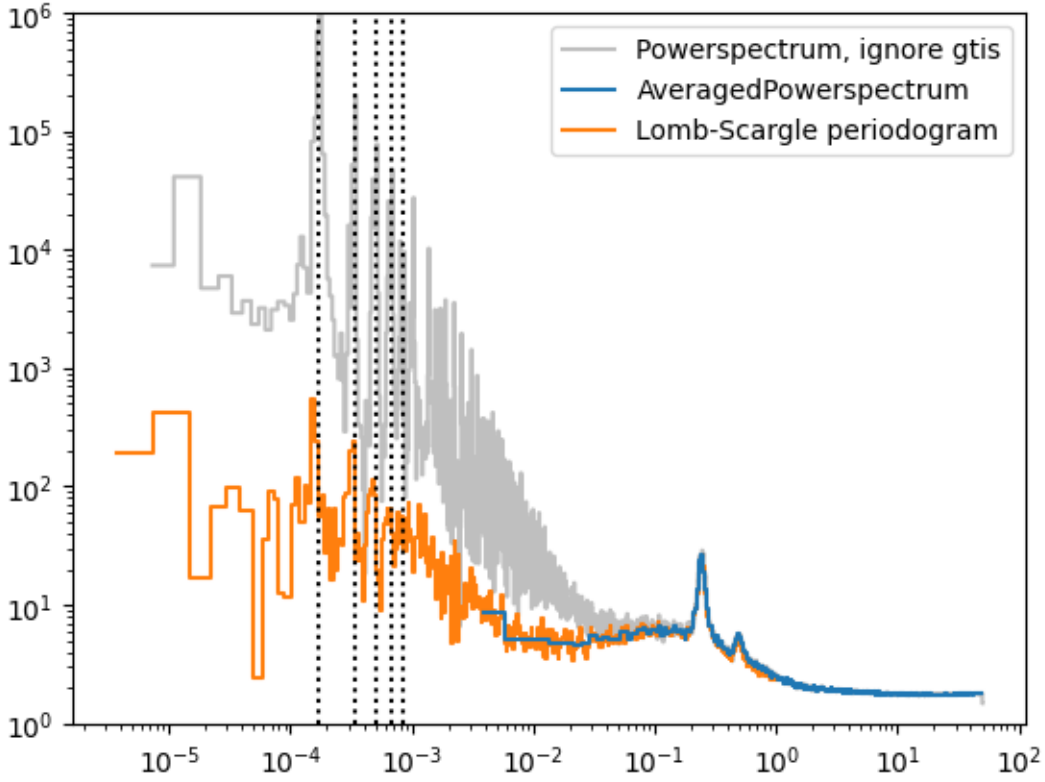
astropy-powered
astropy.org

Timescale, Milestones and KPIs

Timescale	Milestones	KPI
September 2023	Unevenly sampled data sets Optimization	Pull requests #737, #739 (merged) PRs (..) + #755, #756
October 2023	Polarimetric data Single-package Performance test	PR #754 (open) short report (TBD)
April 2024	Scalability tests Optimization planning	Report Report

Accomplished Work, Results – pre-September 2023

- New algorithms for unevenly sampled time series (partly funded by Google Summer of Code 2023):
  - **Lomb Scargle Periodogram and Cross spectrum**
  - Quasi-periodic oscillation modelling with **Gaussian Processes (under review).**
- Code optimization: robust and optimized histogram algorithms
- **Support for polarimetric data (under review)**



Accomplished Work, Results – September 2023

- Contract started on Sept. 1st
- Test conducted on large but smaller than RAM (3 GB) dataset from accreting black hole: comparison with competing (proprietary) code:
  - **Performance**: speed test of many (spectra) timing products: lightcurves, (averaged, single) **periodograms** and cross spectra, lags, coherence (intrinsic, raw)
  - **Robustness**: are results compatible?

- Results:

  - **Performance:** excellent, code generally faster than competitor
  - **Robustness**: good, found one possible bug in the computation of intrinsic coherence.

Next Steps and Expected Results (by next checkpoint: April 2024)

Next steps:

- Finish performance and correctness tests on other spectral-timing products. In particular:
  - Rms spectra, covariance spectra
  - Intrinsic and raw coherence,
  - Time lags
- Scalability test on the code for a data set at least 30 times bigger than the RAM
- Formation (school on **containerization**)
- Strategy to optimise the code

Next Steps and Expected Results (by next checkpoint: April 2024)

We need to make sure that Stingray **scales** correctly to very large datasets

- A **linear** or close to linear scalability of the code for the big data set (in particular, no exponential behavior)
- From the developers, we expect to find a good strategy to optimise the code once found the possible bottlenecks and performed the scalability test