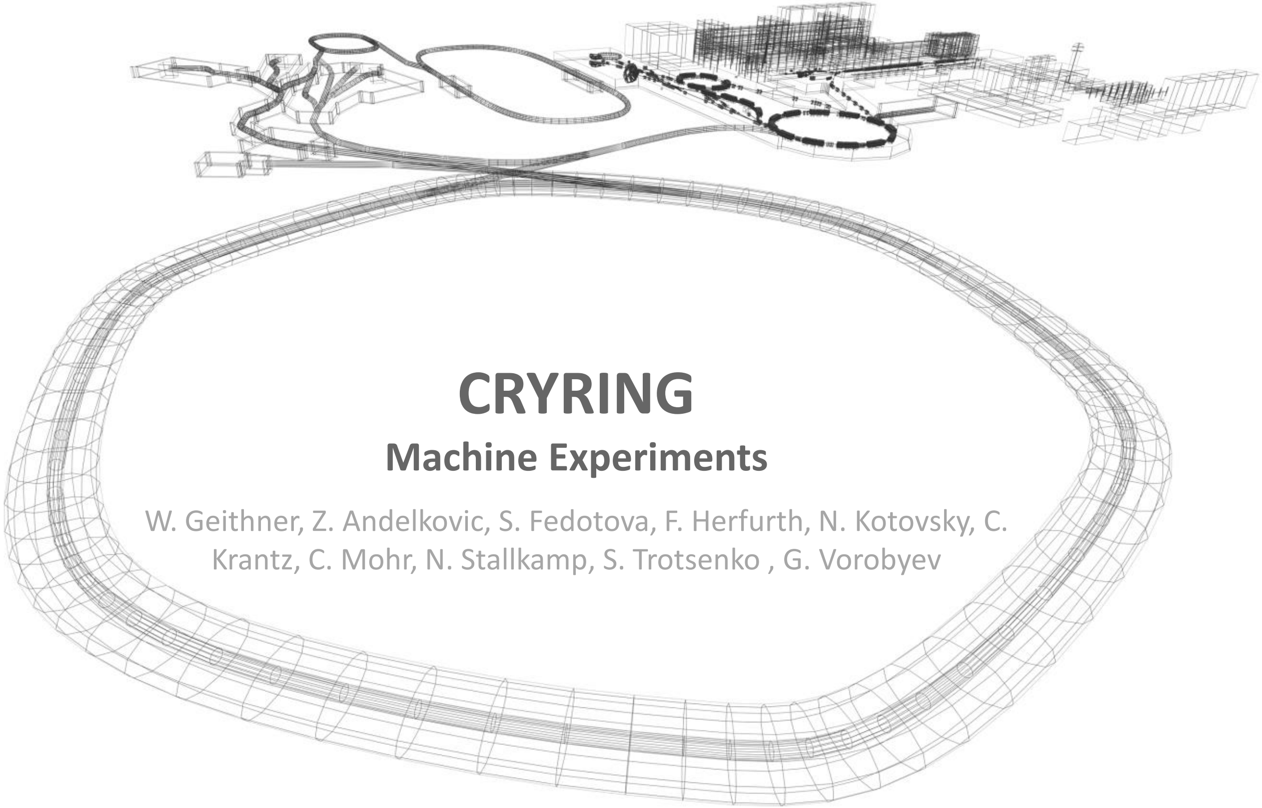


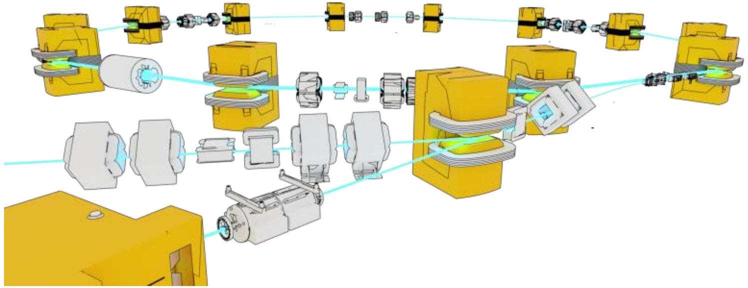
# CRYRING

## Machine Experiments

W. Geithner, Z. Andelkovic, S. Fedotova, F. Herfurth, N. Kotovsky, C. Krantz, C. Mohr, N. Stallkamp, S. Trotsenko , G. Vorobyev

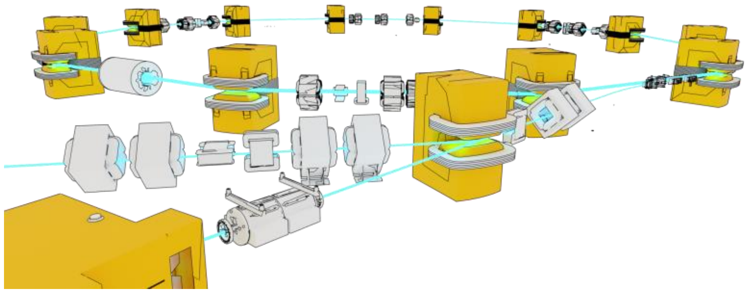
# Machine Experiments 2025

- CRY-AP-255: Multi turn injection optimization
- CRY-AP-256: Schottky detection using BPMs
- CRY-AP-257: Testing the ML and MC optimization routines of DeviceAutomator on injected ion beam
- CRY-AP-258: Working point (tune) and chromaticity measurements using DeviceAutomator
- CRY-AP-259: Test cooling with expansion 10 „only“
- CRY-AP-260: Deceleration from max to min and/or orbit control
- CRY-AP-261: Stacking „in between bunches“ using the B2B RF synchro
- CRY-AP-262: Stacking using the E-Cooler
- CRY-AP-263: Continue working on direct route SIS18 YR : use kicker to inject into YR and measure attainable intensity



# Machine Experiments 2025

- CRY-AP-255: Multi turn injection optimization
  - motorized drive for septa plates and bumper plates installed and used for the first time
  - optimized injection for a number of ion beams, however, still developments needed towards actual understanding
- CRY-AP-256: Schottky detection using BPMs
  - BPM Schottky type signal stronger (due to higher amplification) but also more noise : thinkable of using in place of the dedicated long Schottky pickup
  - Improved amplification (charge-sensitive versus voltage-sensitive) yields a much better Schottky signal even for weak and slow beams (Thanks to BEA!)
- CRY-AP-259: Test cooling with expansion 10 „only“
  - Done. Seems possible, however, sort of defeats the point of CRYRING@ESR
- CRY-AP-260: Deceleration from max to min and/or orbit control
  - First tests done, however, orbit control still needs investigation (software response time).



CRY-AP-257: Testing the ML and MC optimization routines of DeviceAutomator on injected ion beam

next

CRY-AP-258: Working point (tune) and chromaticity measurements using DeviceAutomator

next

# DeviceAutomator

## Recent Activities

Wolfgang Geithner, M. Boschert (H_DA), Z. Andelkovic, F. Herfurth,
J. Ködel, N. Stallkamp, S. Trotsenko, D. Zisis

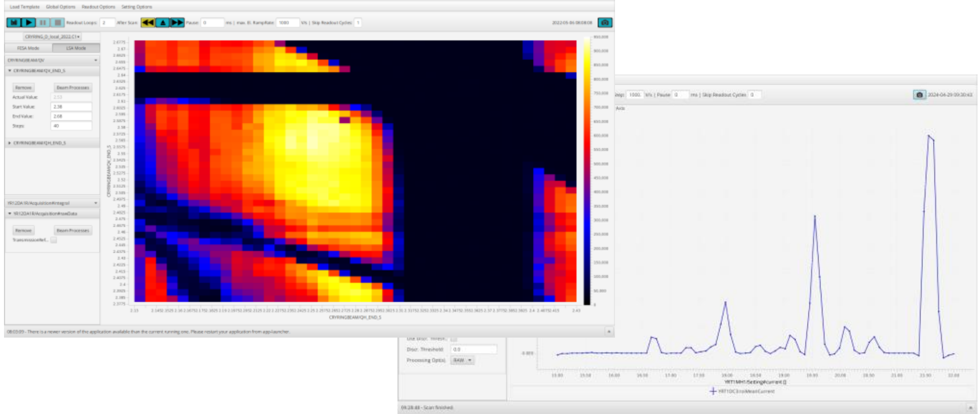
# DeviceAutomator Main Features

- JAVA based Application accessible via Application Launcher
- Goal: Automated parameter setting and coordinated readout of beam diagnostics
- Native integration into FAIR Control System
- Configuration GUI for scan set-up
- „Scan Templates“ for pre-defined scenarios
- Scan algorithms:
  - Bruteforce: parameter stepping at constant step width
  - Genetic optimizer: based on Genetic algorithm
  - Bayesian optimizer: based on Bayesian algorithm
  - Random walk: algorithm perfoming small random walk around local optimum


- Database for configuration and scan data persistence
- Limited to scalar LSA/FESA parameters in the FAIR CS

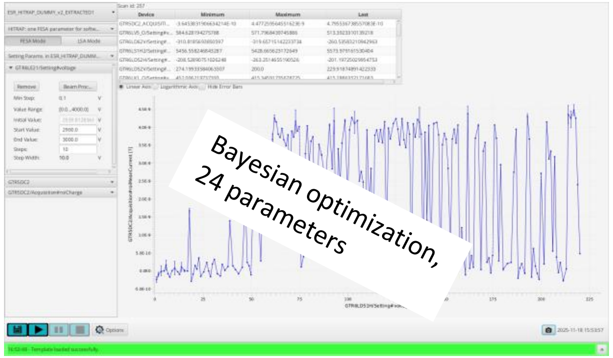
# DeviceAutomator at CRYRING & HITRAP

- CRYRING:
  - Mass spectrum recording
  - Injector/-ion optimization
  - Tune diagram mapping (CRY-AP-258)
- HITRAP
  - Support for manual beamline optimization
  - Deceleration optimization



11/2025:

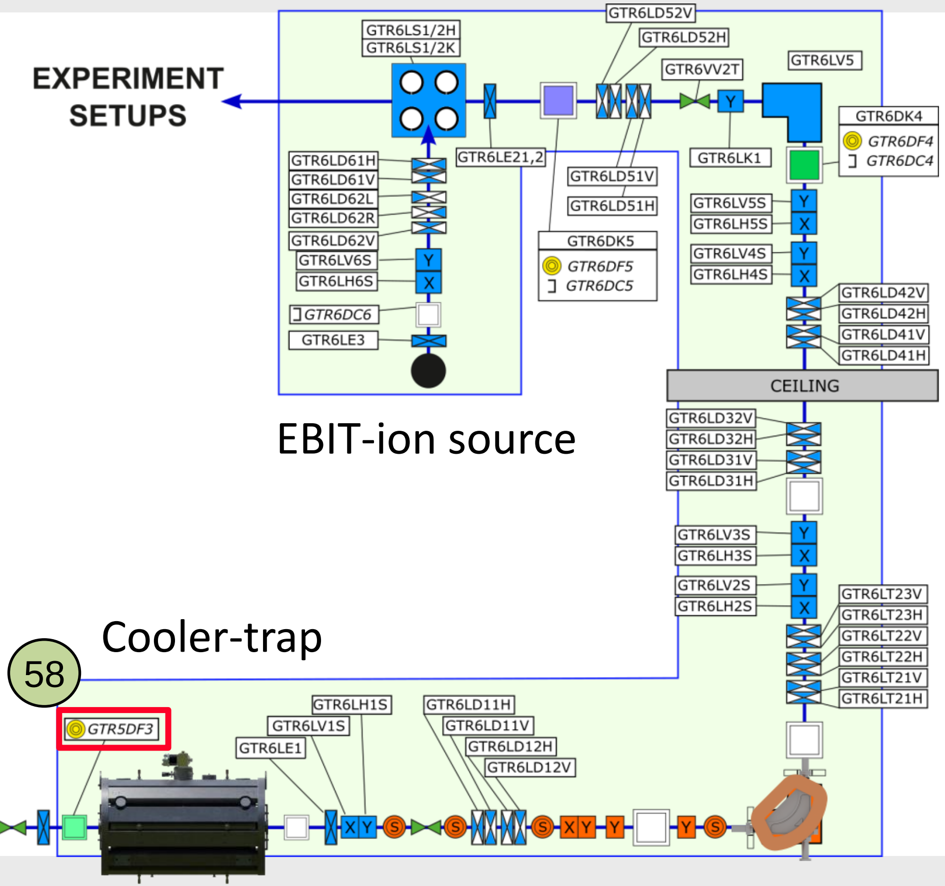
- Focus on optimizers: performance & possible productive use (CRY-AP-257)



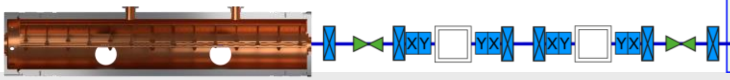
# Beamline Optimization: Challenge

- How to find good beamline settings in 58 parameter space and
  ~ 1000 steps per parameter?

→ Bruteforce with 1 Hz ~ 3 * $10^{166}$ years




RFQ-decelerator



# Very Short Introduction into Optimizer Algorithms in DevAuto

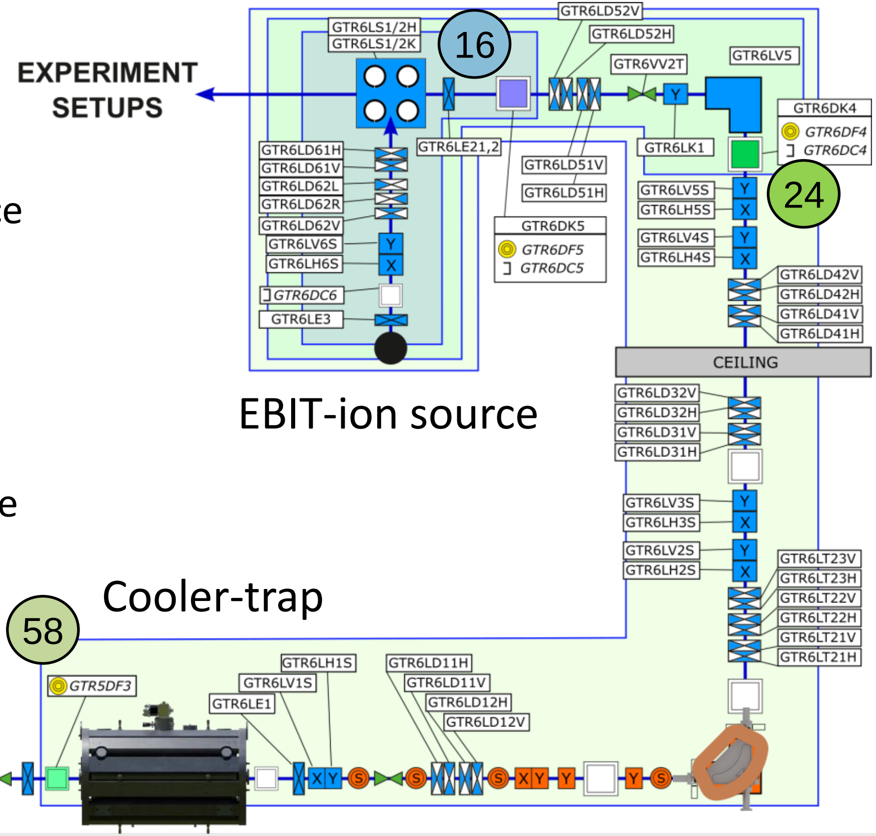
- Genetic Algorithm
  - Based on random values combined with „evolutionary" selection of good values
  - No internal model
  - External JAVA library „Jenetics"
  - DevAuto implementation takes into account power supply stepping resolution
  - Since 2018 used in Device-Automator

- Bayesian Algorithm
  - Compiles internal model of „most promising" function gains for value generation
  - DevAuto uses external JAVA library developed in a Master project at H_DA (informatics department)
  - Power supply resolution not considered in the code
  - Available  since 09/2025

# Experiments at HITRAP – Goals & Environment

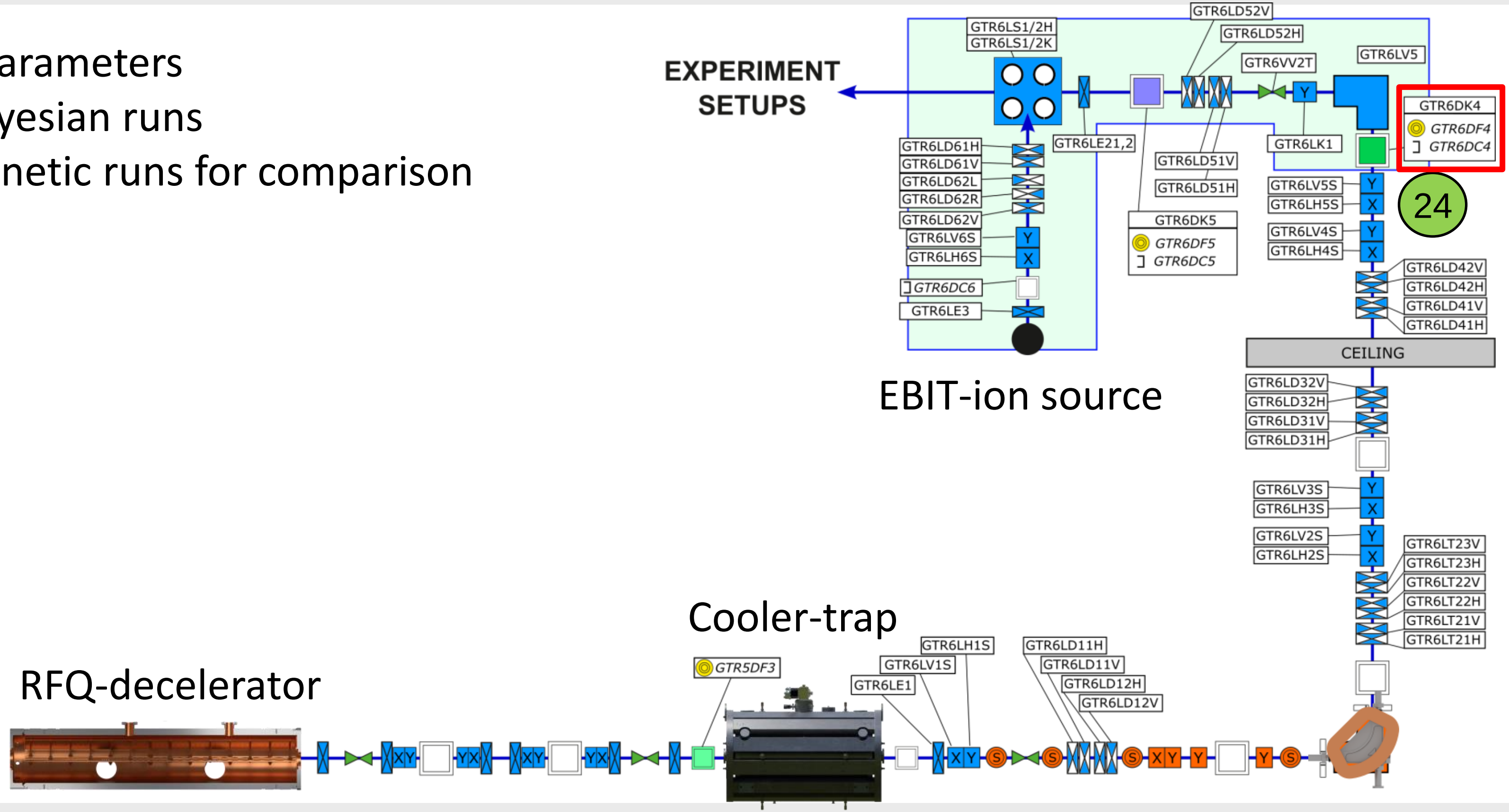
- Integration-test of transition to FAIR CS
- Prepare Bayesian optimizer for standard operation
- Compare performance of Genetic and Bayesian optimizer – convergence speed, quality of convergence
- Collect experience with optimizers
- Check beamline for unexpected behavior

- $^{40}Ar^{16+}$ beam from local EBIT ion source
- Optimization of 16, 24, 58 parameters, single objective
- Parameter ranges ~20% full range




# Experiments at HITRAP: EBIT → GTR6DC4

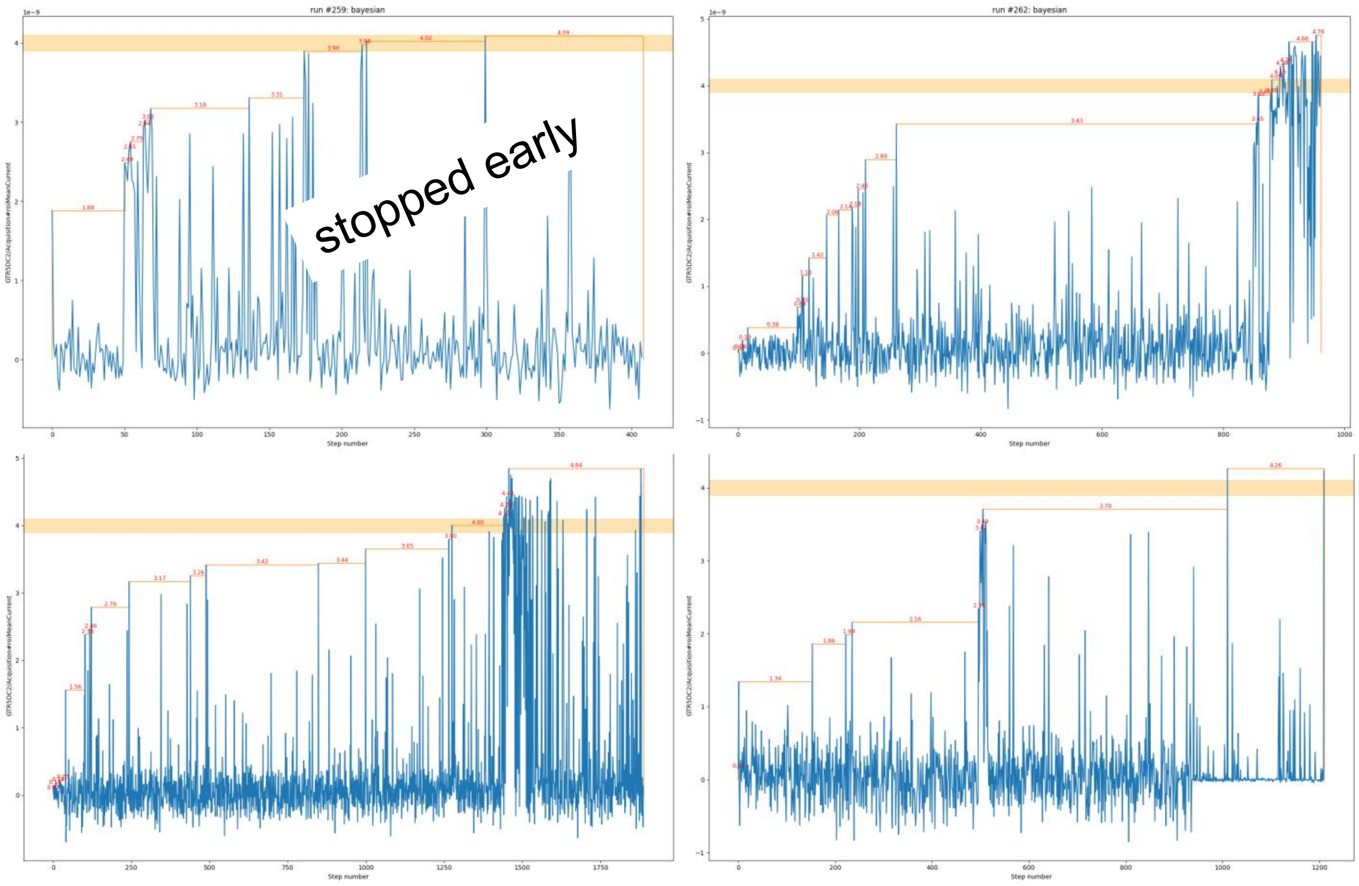
- 24 parameters
- 4 Bayesian runs
- 3 Genetic runs for comparison



# Algorithm tests: Bayesian EBIT -> GTR6DC4

- Mean ROI mean maximum current:
  4.5 nA ~ 115% operator value
- Mean steps to maximum value: 1140
- Mediocre signals reached after ~350-400 steps

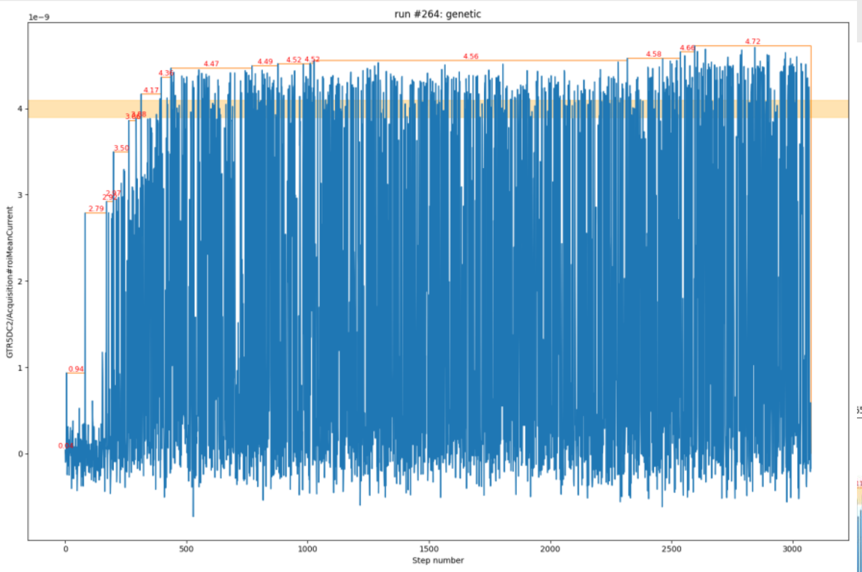
- Sensitive to signal-width vs. parameter-range
- *LogExpectedImprovement* requires re-scaling of objective signals to $\mathcal{O}(1)$
- Internal model impacts performance after ~ 500 steps

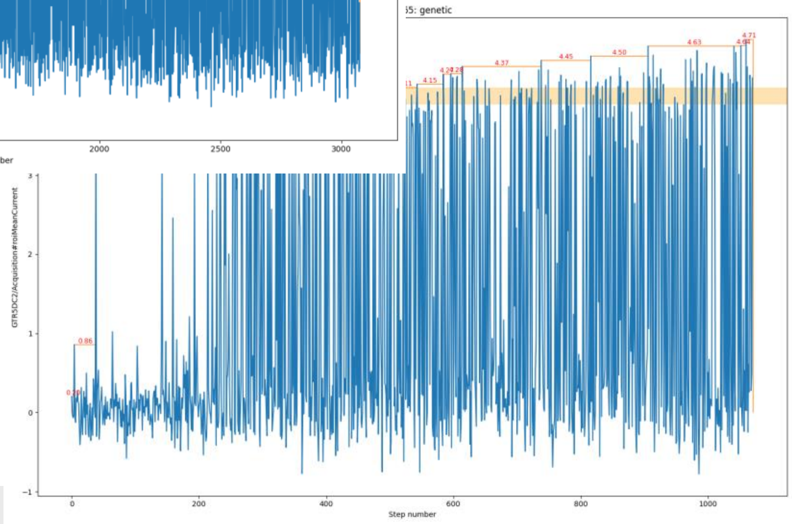


# Algorithm tests: Genetic EBIT -> GTR6DC4

- Average maximum value: 4.7 nA
- Average steps to maximum value: ~1000 steps
- Usable, mediocre signal is reached after averaged 366 steps

- Independent from internal model
  → not (so) sensitive to signal width
- More „random“ than Bayesian (advantageous and disadvantageous)

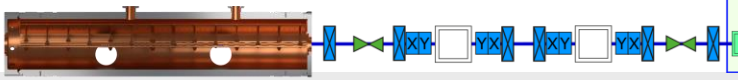


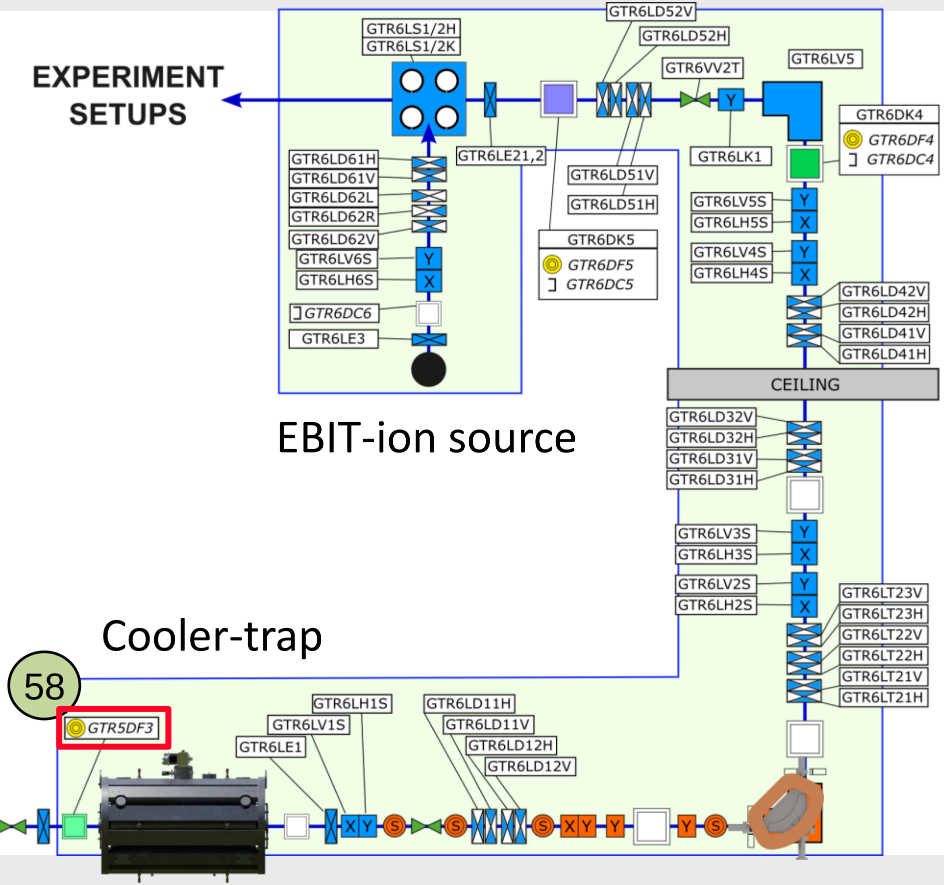



# Algorithm tests: Bayesian EBIT -> GTR5DC2

- 58 parameters
- 3 Genetic runs
- Bayesian runs were tested but did not converge (too large parameter space)

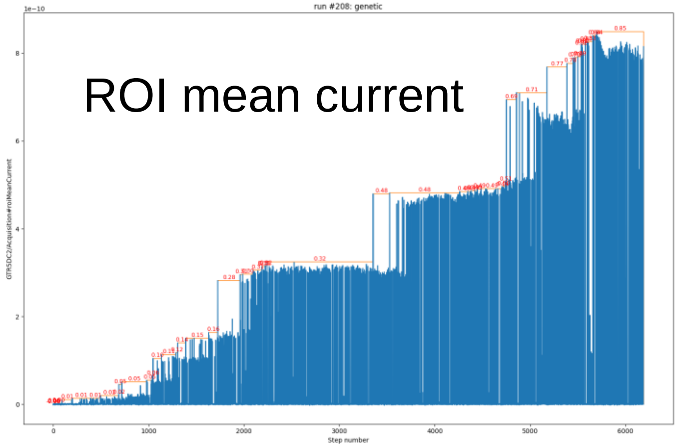
RFQ-decelerator



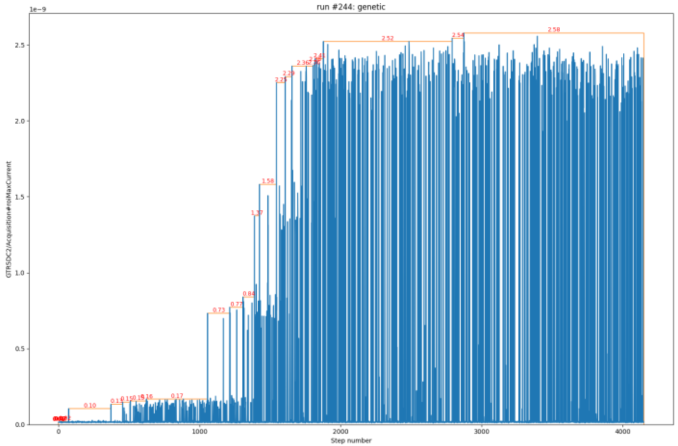


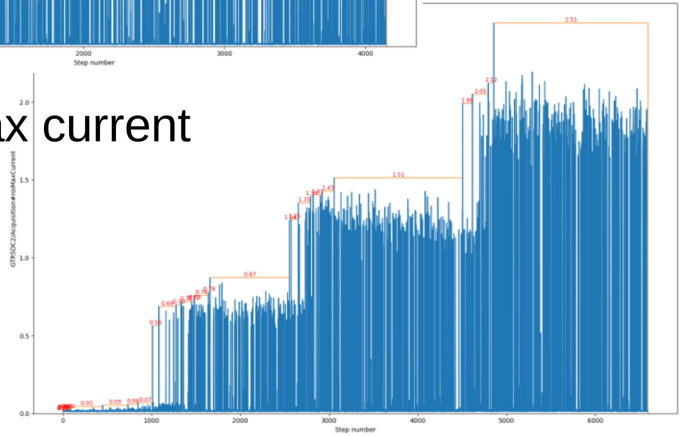
# Algorithm tests: EBIT -> GTR5DC2 Genetic

- Scan performed over up to 18h
- Intensity reached outperformed human operator



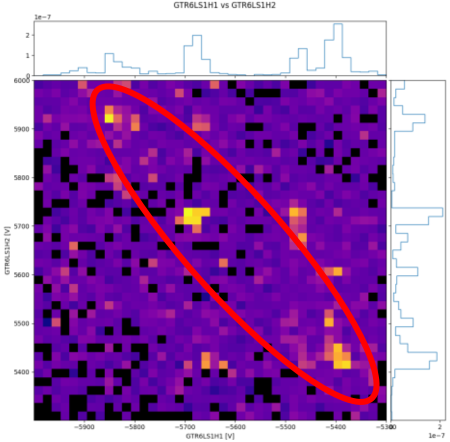

ROI mean current

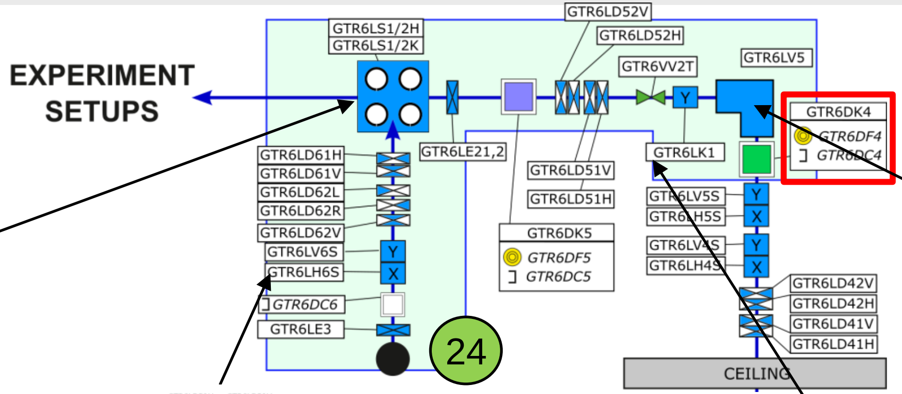


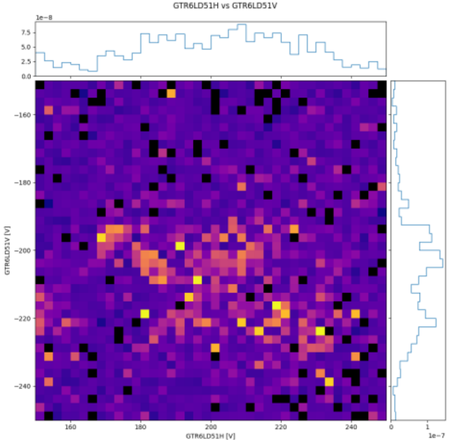



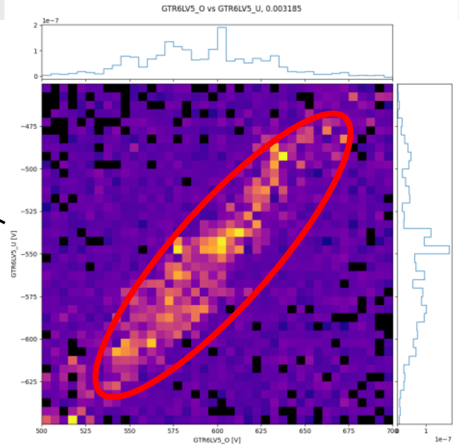

ROI max current

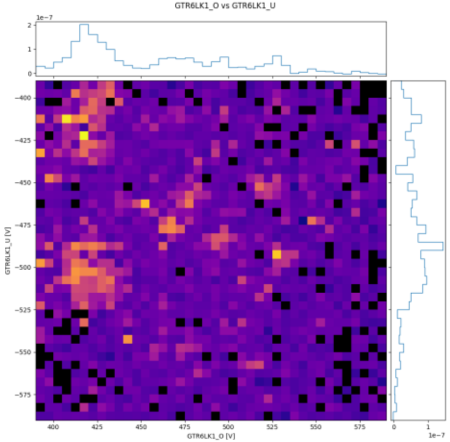
# EBIT → GTR6DC4: Insights from Collected Data



# Conclusions from Optimizer-Tests

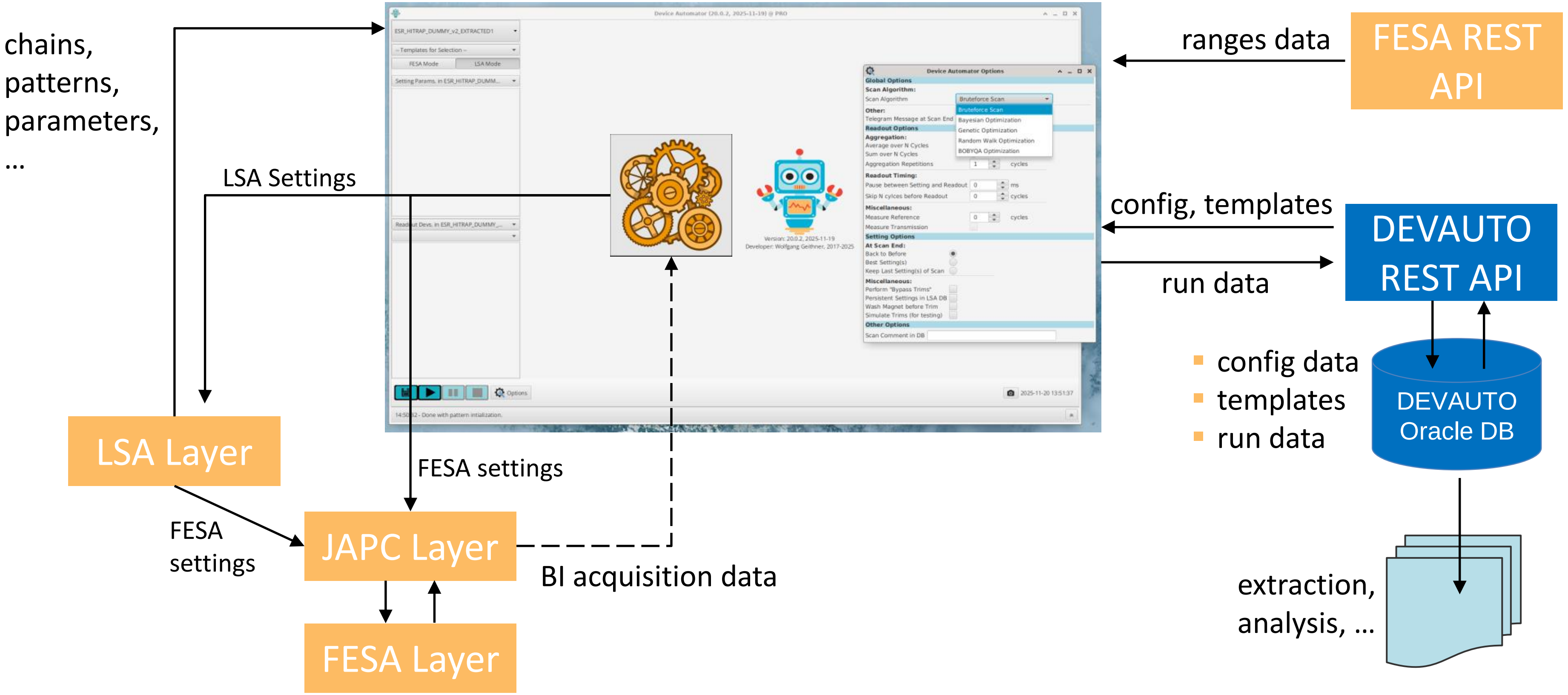
- Optimizers do not outperform human experts

- „Good“ optimizer use cases / scenarios:
  - exploration of unkown (sub)system / beamline settings
  - increase of available time for machine setup
  - tool supporting unexperienced personnel
  - parallel setup of machines
  - limited search spaces
  - Genetic algorithm is better for large search spaces, Bayesian for well known spaces

| Aspect | Genetic | Bayesian |
|---|---|---|
| Reliabiliy of optimum | Good | Good |
| Scan space: parameters and/or ranges | Better for large spaces | Better for narrower ranges |
| Scan Time | Rather random | Faster for narrower ranges |
| Objective range | Independent of objective range | Works better if objective is re-scaled e.g. 1.23E-9 → 1.23 |
| Influence of algorithm on stepping speed | Negligible, constant | Increasing, significant for steps > 1000 |

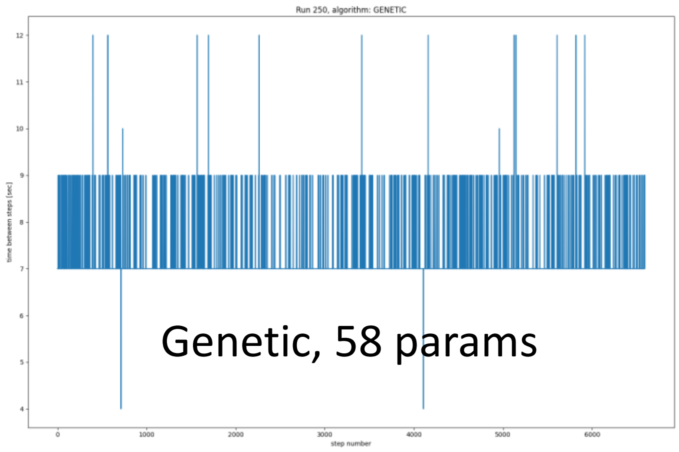
# BACKUP

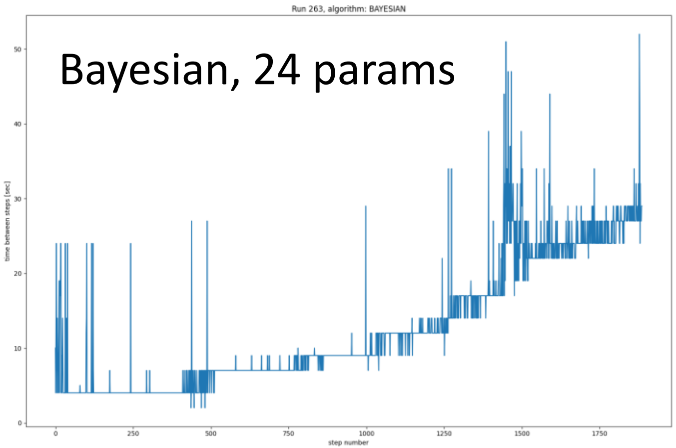
# DeviceAutomator – a bit of Architecture

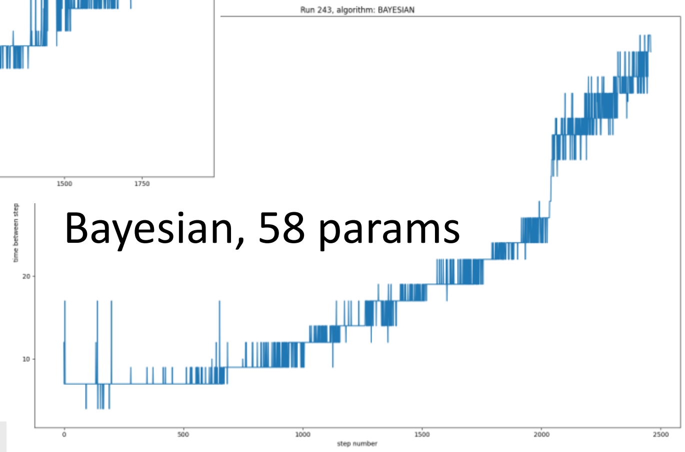


# Bayesian Runs: Algorithm Performance Limits

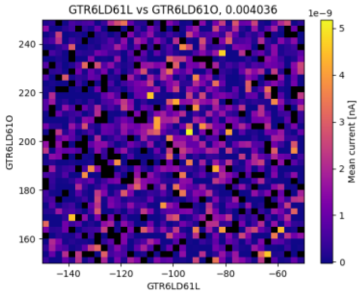
- Genetic algorithm does not do internal calculations $\rightarrow$ constant step-to-step periodicity
- Bayesian algorithm slows down due to internal model calculations

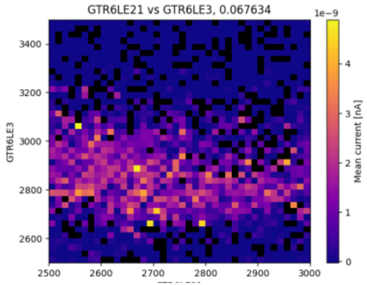


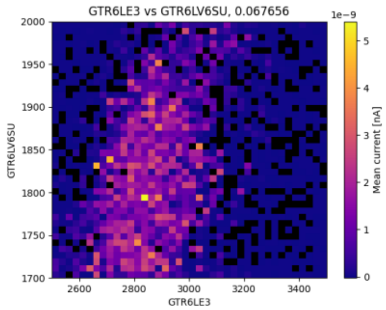



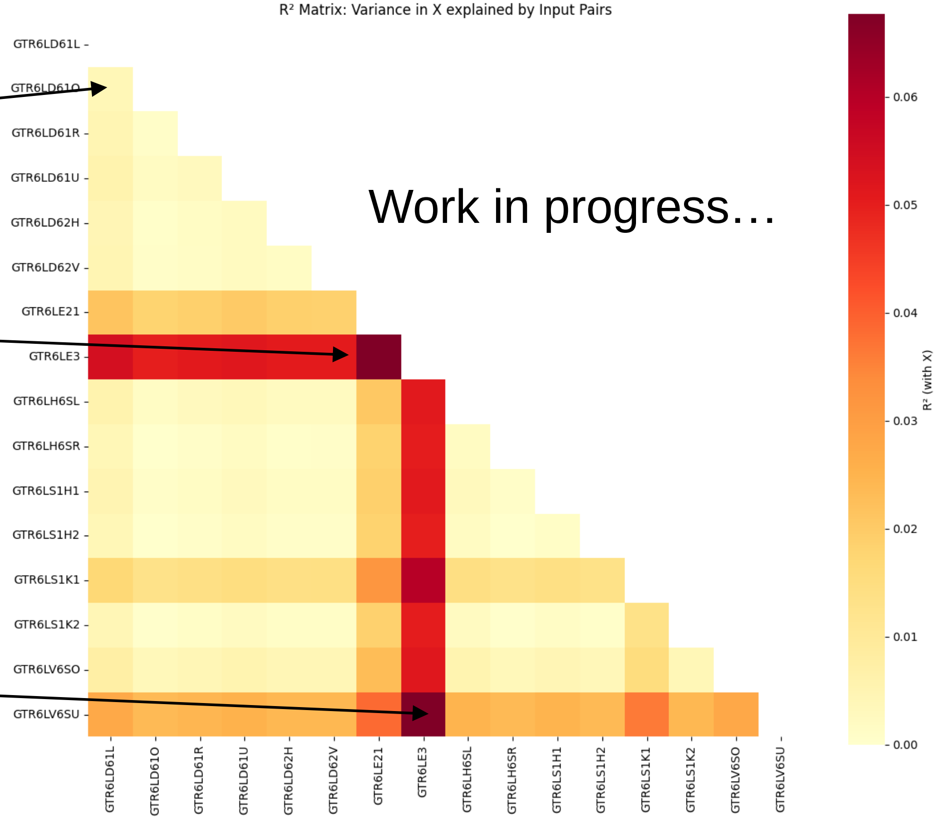



# Tool for Mapping Parameter Ranges / Sensitivity




Work in progress…