[p4] Optimal algorithm to determine pointing centers for PFS open-use programs

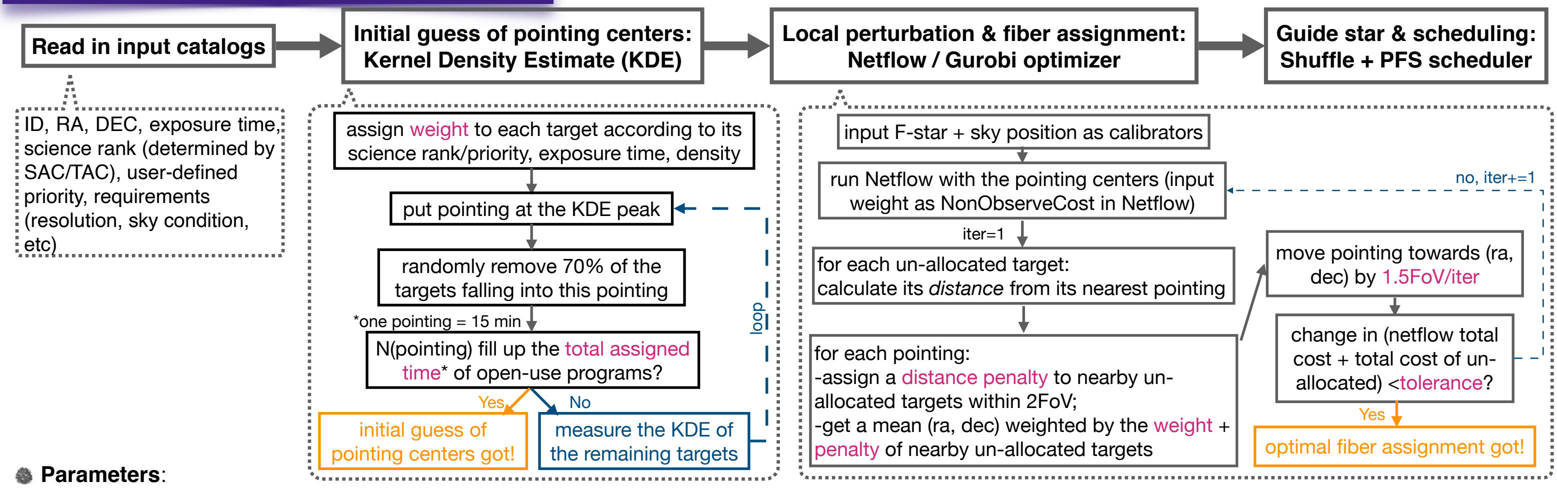
Wanqiu He, Masayuki Tanaka, Miho N. Ishigaki, Masato Onodera (NAOJ) & obsproc member

Introduction

Subaru Prime Focus Spectrograph (PFS) is a fiber-fed multiplex system, which enables acquisition of around 2400 spectra of objects simultaneously over a wide hexagonal field of 1.38 deg on the sky. Its innovative features are expected to help make great improvements in various science fields.

- **Operation Strategy**: to **share fibers among multiple open-use programs** in order to efficiently utilize all the fibers
- **Problem**: where to point the telescope?
  - **targets** from different programs can **vary significantly in spatial density/distribution, required exposure time, science priority,** etc
- In this project, we are going to develop an **"optimal" tiling algorithm for PFS open-use programs**:
  - to achieve high completeness in rank-A (highest science rank) samples
  - to maximize fiber allocation efficiency in each exposure (minimize wasted time)

PFS pointing Planner: a general flow-chart



Read in input catalogs

ID, RA, DEC, exposure time, science rank (determined by SAC/TAC), user-defined priority, requirements (resolution, sky condition, etc)

Initial guess of pointing centers: Kernel Density Estimate (KDE)

assign weight to each target according to its science rank/priority, exposure time, density

put pointing at the KDE peak

randomly remove 70% of the targets falling into this pointing

*one pointing = 15 min

N(pointing) fill up the total assigned time* of open-use programs?

Yes: initial guess of pointing centers got!

No: measure the KDE of the remaining targets

loop

Local perturbation & fiber assignment: Netflow / Gurobi optimizer

input F-star + sky position as calibrators

run Netflow with the pointing centers (input weight as NonObserveCost in Netflow)

iter=1

for each un-allocated target: calculate its *distance* from its nearest pointing

for each pointing:
-assign a distance penalty to nearby un-allocated targets within 2FoV;
-get a mean (ra, dec) weighted by the weight + penalty of nearby un-allocated targets

move pointing towards (ra, dec) by 1.5FoV/iter

change in (netflow total cost + total cost of un-allocated) <tolerance?

no, iter+=1

Yes: optimal fiber assignment got!

Guide star & scheduling: Shuffle + PFS scheduler

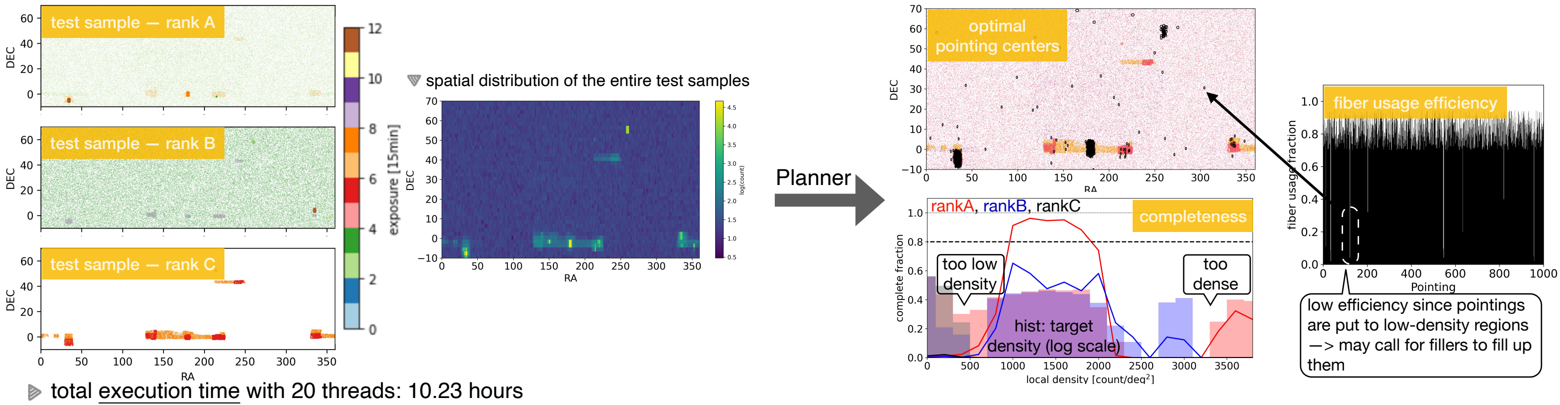
- **Parameters**:
  - total time allocated to open-use programs —> determined by SAC/TAC
  - weight: give **higher** weights to samples with **rank A** (highest science rank)**;** additional weights might be added to samples with shorter exposure time and higher local density (finish samples requiring short exposure time as soon as possible)

W1	**science rank**	**Rank A - P=3; Rank B - P=2; Rank C - P=1**	**W1=pow(a, P)**
W2	**exposure**	**N(exposure frame)**	**W2=pow(N, b)**
W3	**local density**	**N(target) in FoV**	**W3=pow(N, c)**

  - distance penalty: give **larger** penalty to targets **far** away from the pointing center —> prevent moving pointing too far away from its initial guess
  - combination between weight and distance penalty, moving step, tolerance, PA of each pointing, etc. —> optimizing all the parameters is still in progress…
- **Outputs**: pointing list + fiber assignment in each pointing

Simulation test: an example

- simulate 1000 pointings (one pointing=15min) over a set of test samples (weight scheme: a=10, b=-0.2, c=0)



low efficiency since pointings are put to low-density regions —> may call for fillers to fill up them

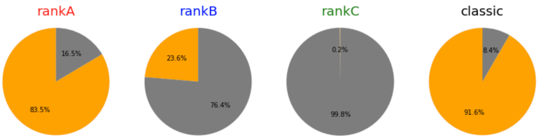
- total execution time with 20 threads: 10.23 hours
- complete fraction: **rank-A samples** in a proper range of spatial density can **achieve a completeness of >90%**
  - due to the fixed density of fibers (2394/FoV), samples with too-low or too-high density are hard to achieve a high completeness within the allocated time
  - PFS (queue mode) might not be suitable for isolated samples with too-low density or too-dense samples
- fiber usage efficiency: **72.2%** fibers are used on average

PFS pointing Planner: online version

- a **simplified version** of Planner can be provided to help users get an idea of expected completeness of their science samples
  - **one example:** user's science sample —> ~1900 per sq.deg, covering ~4 sq.deg, request 1 hour per target

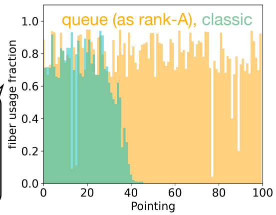
output: expected completeness if the user get rankA/rankB/rankC score in the queue mode or 5 nights (maximum for one program) in the classic mode

orange fraction: completed fraction



if user can get rank-A, the expected completeness can be as good as that got by the classic mode

usage of fibers is more efficient in the queue mode than the classic mode



Subaru UM FY2021, Jan. 11-13, NAOJ