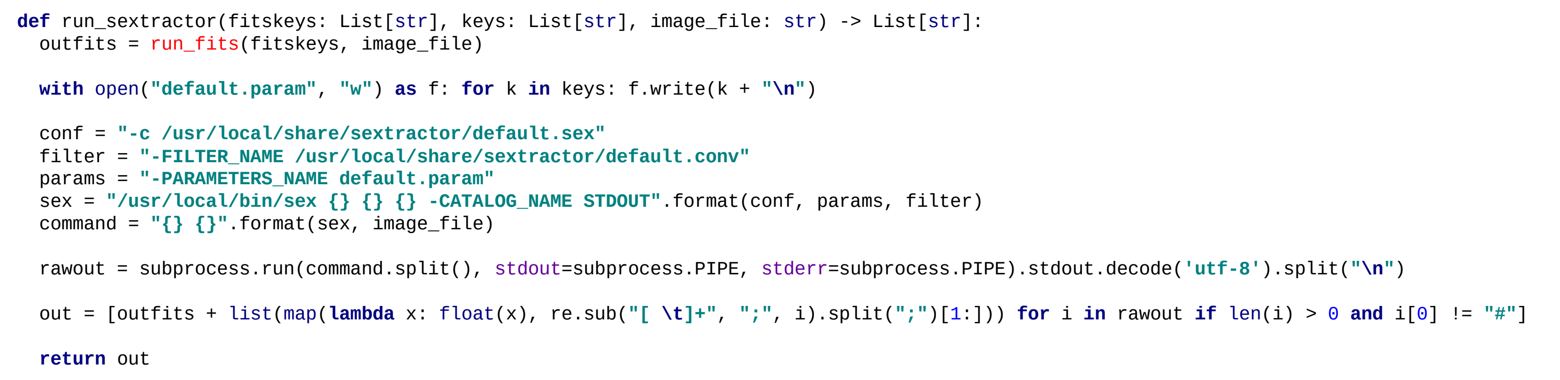
Sextractor & Spark

- https://www.astromatic.net/software/sextractor
- Installed in the Spark cluster @LAL (all workers) (+ lapack, atlas, fftw)
- Principle to install sextractor into a Spark pipeline:
  - Sextractor is a standalone application to be applied onto one single FITS image
    - By default a set of parameters are provided in the sextractor package (=> default algorithms)
    - Extract objects from the image into a catalog (=> a table of object characteristics [float])
    - Object characteristics must be selected from a fixed list of ~300 predefined keys => flot values
  - => we parallelize N runs from N images
  - Output of a run is sent to the Spark data flow as one combined dataframe
    - Can select values extracted from the FITS header file
    - Send all selected catalog columns
  - Then the assembled dataframe may be used to perform combined analyses

Sextractor & Spark

- To submit one sextractor run, we define one UDF
  - Input:
    - The FITS file name
    - List of FITS keys to be extracted from the FITS header: List[any]
    - list of selected keys to the catalog production : List[float]
  - Operation:
    - read the FITS header
    - Build the **`param.default`** file to be given to sextractor
    - Run sextractor and format catalog from stdout
  - Ouput: (the selected header values + the produced catalog) : List[any]



```python
def run_sextractor(fitskeys: List[str], keys: List[str], image_file: str) -> List[str]:
  outfits = run_fits(fitskeys, image_file)

  with open("default.param", "w") as f: for k in keys: f.write(k + "\n")

  conf = "-c /usr/local/share/sextractor/default.sex"
  filter = "-FILTER_NAME /usr/local/share/sextractor/default.conv"
  params = "-PARAMETERS_NAME default.param"
  sex = "/usr/local/bin/sex {} {} {} -CATALOG_NAME STDOUT".format(conf, params, filter)
  command = "{} {}".format(sex, image_file)

  rawout = subprocess.run(command.split(), stdout=subprocess.PIPE, stderr=subprocess.PIPE).stdout.decode('utf-8').split("\n")

  out = [outfits + list(map(lambda x: float(x), re.sub("[ \t]+", ";", i).split(";")[1:])) for i in rawout if len(i) > 0 and i[0] != "#"]

  return out
```

Sextractor & Spark

- The Spark pipeline

```
fitskeys = ["RA_DEG", "DEC_DEG", "FILTER", "RUNID"]
keys = ["NUMBER", "EXT_NUMBER", "FLUX_ISO", "MAG_ISO", "ALPHA_SKY", "DELTA_SKY", "FLUXERR_ISO", "MAGERR_ISO"]

images_for_CFHT_dataset = "/lsst/data/CFHT/rawDownload/*/*.fits"
files = random.sample(glob.glob(images_for_CFHT_dataset), N)

rdd = spark.sparkContext.parallelize(files, len(files)).flatMap( lambda x : run_sextractor(fitskeys, keys, x))

df = rdd.toDF(fitskeys + keys).cache()
data = df.sample(False, 0.1).select("ALPHA_SKY", "DELTA_SKY").toPandas().get_values().transpose()

x = data[0].astype(float)
y = data[1].astype(float)
plt.scatter(x, y, marker='.')
plt.show()
```