

```

"""
Script used to match the traces from different observers.
"""

from itertools import combinations

import pandas as pd
import numpy as np
from scipy.optimize import linear_sum_assignment

year = 2018

# Loading data from an Excel file. A DataFrame is generated for each
# observer.
df = pd.read_excel(f"Tab1_{year}.xlsx")
df = df[2:]

df.loc[:, "Well_ID"] = df.loc[:, "Well_ID"].ffill()

properties = ["number", "degree", "length"]

obs = (len(df.columns) - 1) // len(properties)

new_columns = ["well_id", *properties]

obs_dfs = []
for i in range(obs):
    # Extract obs i
    index = np.arange(len(properties)) + i * len(properties) + 1
    obs_df = df.iloc[:, [0, *index]]

    # Rename columns
    obs_df = obs_df.rename(
        columns={old: new for old, new in zip(obs_df.columns, new_columns)}
    )
    obs_df = obs_df.drop(columns=["number"])

    # Handle degrees
    obs_df["degree"] = obs_df["degree"] != "_"
    obs_df["degree"] = obs_df["degree"].notna()
    obs_df["degree"] = obs_df["degree"].str.replace("N", "").astype(int)

    # Add obs_id

```

```

obs_df.insert(1, "obs_id", i)

obs_dfs.append(obs_df)

# Create unique id for each trace per well
all_obs = pd.concat(obs_dfs, ignore_index=True)
all_obs["track_id"] = all_obs.groupby("well_id").cumcount()
dfs = {obs: group for obs, group in all_obs.groupby(all_obs["obs_id"])}

# Matching the traces between observers

def diff(a1, a2):
    diff = abs((a1 - a2) % 360)
    return min(diff, 360 - diff)

def match_obs(obs_1, obs_2):
    common_wells = set(obs_1["well_id"]) & set(obs_2["well_id"])
    for well_id in common_wells:
        df1 = obs_1[obs_1["well_id"] == well_id]
        df2 = obs_2[obs_2["well_id"] == well_id]

        # linear sum assignment
        matrix = np.array(
            [
                [diff(a1, a2) for a2 in np.array(df2["degree"])]
                for a1 in np.array(df1["degree"])
            ]
        )
        matrix[matrix > 20] = -10000 # distance max between two tracks is
20°

        lines, columns = linear_sum_assignment(-matrix)
        indexes = [[l, c] for l, c in zip(lines, columns) if matrix[l, c] >=
0]

        # update obs_2 track_id
        for i1, i2 in indexes:
            obs_2.at[df2.index[i2], "track_id"] = df1.at[df1.index[i1],
"track_id"]

    for obs_1, obs_2 in combinations(dfs.values(), 2):
        match_obs(obs_1, obs_2)

```

```
df_final = pd.concat(dfs.values())  
df_final = df_final.sort_values(by=["well_id", "track_id"])  
df_final.to_excel(f"Tab2_{year}.xlsx")
```