**Using the ‘Matching\_Package’**

**By Michael Khela (michael.khela99@gmail.com) on 6/9/2024**

BackgrounD

This protocol provides instructions/documentation for **using the ‘Matching\_Package’ to create pairs of IDs that are matched based on a numerical matching column and gender, if applicable.**

Needed resources and/or files:

1. Matching\_Package (Found on the Wilkinson Lab google drive)
2. A CSV that contains the following columns: ID, cohort, matching, and sex (if applicable)
3. The script was built and has been tested using Python (version 3.12.1) and the following packages:
   1. – pandas (version 2.2.0)
   2. - openpyxl (version 3.1.2)
4. You can install these packages using pip: “pip install pandas openpyxl” in the python terminal

## **Important File Paths**

**Matching\_Package (folder on g drive)**

/Wilkinson\_Lab/Analysis Tools/Participant Matching/Matching\_Package

**NOTE: You will NEED to copy this folder to a destination that you will run this script in. After doing this, you must specify destination of the package in “root\_filepath” in the script**

1. Some broad context about the Matching package

1. The Matching\_Package was created to optimize the process creating matches between participants. Rather than an individual doing it by hand, the code takes in the users input and outputs a matched file in which the participants are matched.
2. The package needs one files as input:
   1. A CSV containing the data that will be matched
3. The package outputs one file:
   1. Matching\_Results\_YYYY-MM-DD\_[number].csv
      1. **NOTE: The [number] is dynamic. It will autodetect if a file has existed and will add a suffix**

2. Running Group\_Matching.py

1. Open Group\_Matching.py by double clicking it

A blue line on a black background

Description automatically generated

1. On the code, edit the following lines (as indicated on the script):

A screenshot of a computer

Description automatically generated

* 1. DO NOT EDIT anything else
  2. At the end of your file path to the ‘Matching\_Package’ folder, **add a forward slash (/)**
  3. Set ‘comparison\_group’ to the groups **you want to compare**, separated by commas.
  4. Set ‘control\_group’ to the control group for **comparison**.
     1. **NOTE**: the control will be compared to EACH of the comparison groups once. (i.e. control\_1 can be matched for compare\_1, compare\_2, etc)
  5. ‘max\_match\_diff’: Define the maximum allowable difference in the numerical matching column between groups.
  6. ‘id\_column’, ‘cohort\_column’, ‘matching\_column’, ‘sex\_column’: Set these to the respective column names in your input file. If ‘sex\_column’ is not needed, set it to ‘None’.
  7. **NOTE:**
     1. Ensure the matching column contains numerical values.
     2. If gender-based matching is not required, set `sex\_column` to `None`.
     3. Review the output file to verify the matching results.

1. Run the script

A screenshot of a computer

Description automatically generated

1. You’ll know the script is done running when you see the message “Matched Report is Created!!” and then the output file!
2. You’ll find the **output file in the ‘Output’ folder in the package**

In case of errors:

1. If the code shows any error message, please take a screenshot and send it to **Michael Khela or Carol Wilkinson.**
2. If the script fails to read the input file, check the file path and ensure the file extension is correct.
3. Ensure column names in the configuration match those in the input file exactly.