

# AME: Interpretable Almost Exact Matching for Causal Inference

**Haoning Jiang**

**Thomas Howell**

**Neha R. Gupta**

**Vittorio Orlandi**

**Marco Morucci**

**Harsh Parikh**

**Sudeepa Roy**

**Cynthia Rudin**

**Alexander Volfovsky**

*Duke University*

HJ112@DUKE.EDU

THOMAS.HOWELL@DUKE.EDU

NEHA.R.GUPTA@DUKE.EDU

VITTORIO.ORLANDI@DUKE.EDU

MARCO.MORUCCI@NYU.EDU

HARSH.PARIKH@DUKE.EDU

SUDEEPA@CS.DUKE.EDU

CYNTHIA@CS.DUKE.EDU

ALEXANDER.VOLFOVSKY@DUKE.EDU

**Editors:** Douwe Kiela, Marco Ciccone, Barbara Caputo

## Abstract

**AME-GUI** (**A**lmost **M**atching **E**xactly **G**raphical **U**ser **I**nterface) is an interactive web-based application that allows users to perform *matching* for *causal inference* on large, complex datasets with categorical covariates. The application is powered by the *Fast Large-Scale Almost Matching Exactly (FLAME)* algorithm (Wang et al., 2021), which matches treatment and control units in a way that is (i) *interpretable*, because the matches are made directly on covariates, (ii) *high-quality*, because machine learning is used to determine which covariates are important to match on, and (iii) *scalable*, using techniques from data management. The graphical user interface highlights the utility of this algorithm and uses a suite of visualization tools to facilitate easy and interactive exploration of treatment effect estimates, as well as of the created matched groups that they depend on. The application gives a quick and simple overview of the open-source Python and R packages **dame-flame** and **FLAME**, and the range of functionality they provide for interpretable and efficient causal inference.

**Keywords:** Causal Inference, Matching, Observational Study, Interpretability, Graphical User Interface

## 1. Introduction

We present **AME-GUI**<sup>1</sup>, a web-based graphical user interface (GUI) for *observational causal analysis* using **A**lmost **M**atching **E**xactly algorithms. These algorithms, under certain assumptions, provide valid causal estimates from observational studies that are alternatives to randomized controlled trials when the latter cannot be performed due to logistical or ethical reasons (Rubin, 2005; Rosenbaum and Rubin, 1983). In an observational study, it is likely that those who are treated (e.g. taking a drug or participating in a job training program) and those who are not treated, are somewhat dissimilar. However, when asking causal questions, we are interested in contrasting outcomes of a population had everyone

1. <https://almost-matching-exactly.github.io/ame-demo>

been treated and outcomes had no one been treated. To do so, we compare the outcomes of units with different treatment statuses and must take care to disentangle differences due to treatment (indicative of a causal effect) and those due to differences in other features (indicative of confounding). **AME-GUI** allows users to quickly and easily accomplish this via *matching*. Matching methods identify individuals with opposite treatments that can be compared to one another to estimate treatment effects because they are similar in many other respects. Matching is an inherently interpretable method for observational causal inference as it allows the users to audit the exact data that led to causal effect estimates by examining matched groups. **AME-GUI** offers a range of customizable parameters and includes built-in plotting tools to view results of the matching and the corresponding effect estimates. The GUI is built using Flask with a Python backend and relies on the open source Python package **dame-flame** (Gupta, 2020). It is hosted live online and requires no special downloads or set-up to access.

## 2. The Algorithm

**AME-GUI** is powered by matching algorithms developed by the Almost Matching Exactly lab at Duke ([AME-Lab](#)), and in particular uses the *FLAME (Fast Large-scale Almost Matching Exactly)* algorithm (Wang et al., 2021). The AME framework operates on the principle that units should be matched together if they are close in their covariates according to an appropriate metric. Because high dimensional settings make it unlikely that units are similar in *all* their covariates, matches must be made on a *subset* of all possible covariates. Humans are not naturally adept at the task of deciding which high dimensional covariates one should match on, so FLAME uses machine learning on a holdout dataset to learn measures of covariate importance and thereby identify high quality subsets. Assumptions standard in observational causal inference must be made in order for the FLAME algorithm to be applicable. These include the *Stable Unit Treatment Value Assumption (SUTVA)* (there is only one version of treatment and treatments applied to one unit do not affect the outcomes of other units), and the conditional *unconfoundedness* or *ignorability* assumption (the outcome is independent of the treatment assignment conditional on covariates).

The FLAME algorithm operates on an input dataset containing categorical covariates, a binary treatment, and a continuous or categorical outcome. It begins by matching any units that can be matched exactly on all covariates. Units that have identical values form a *matched group*, as long as the group contains at least one treatment and one control unit. The algorithm then iterates: it drops a single covariate, creates matched groups from any units matching exactly on remaining covariates, and repeats until a stopping criterion is reached. To determine the best covariate to drop at each iteration, FLAME decides to match on the covariates yielding the highest *match quality* MQ, defined as  $MQ = C \cdot BF - PE$ , where  $C$  is a user-specified hyperparameter. The term PE denotes the *predictive error*, which estimates the error associated with not using the dropped covariates to predict the outcome. This is done via a user-chosen machine learning algorithm that is run on a *holdout training set*. The *balancing factor*, BF, measures the proportion of treatment and control units that are matched on a covariate set. Pseudocode and additional details about the FLAME algorithm can be found in Wang et al. (2021).

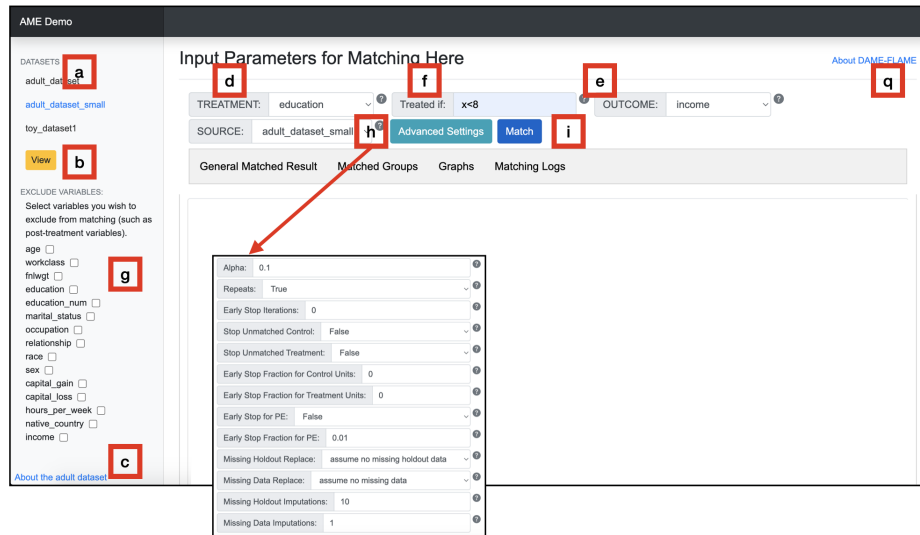


Figure 1: Snapshot of the AME GUI

### 3. The GUI

In this section, we briefly discuss the GUI for **AME-GUI**; a snapshot is displayed in Figure 1. The GUI incorporates inputs from users, and provides detailed insights into the matching process and causal effect estimates returned by the algorithm.

**Inputs for Matching.** First, users can choose from the available datasets [a] and preview the contents through the “View” button [b]. The desired treatment [d] and outcome [e] variables can be selected from menus. The user has a few options for cleaning their dataset prior to running the matching procedure. If the treatment variable is not already encoded as 0 or 1, as required, the user can specify a predicate [f] that encodes the treatment of any complying units as 1, and that of any noncomplying units as 0. Users can also exclude variables, such as post-treatment variables, from the matching procedure [g]; doing so will not permanently remove them from the dataset.

The user has access to several advanced settings [h] that tune the matching algorithm. For example, users can handle missing data by either excluding units with missingness from the procedure or by imputing the missing data via Multiple Imputation by Chained Equations (MICE) (Buuren and Groothuis-Oudshoorn, 2010). Users are also given a number of options for determining stopping conditions for the algorithm. The algorithms must stop when all units are placed in matched groups or all covariates have been dropped, but other options are available, such as stopping when there are too few unmatched (treatment or control) units or after a certain number of iterations. Users can also choose whether a previously matched unit can be matched in a subsequent iteration of the algorithm. After all settings have been selected, the user will click the “Match” button [i] to run the algorithm.

**Outputs from Matching.** Upon matching (Figure 2), the GUI displays a table containing all the matched units [j]. This table includes all matched units and indicates which covariates each unit was matched on; covariates that were not used for matching are indicated with an asterisk (\*) in place of their values. The estimated *average treatment effect*

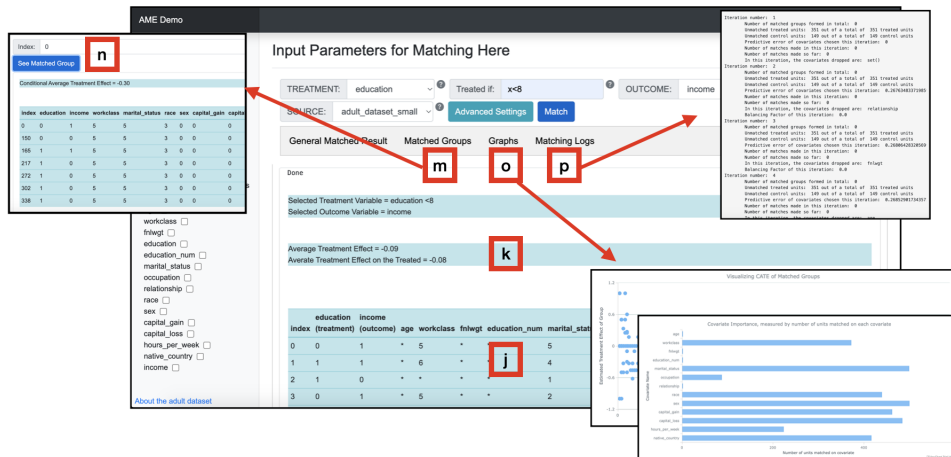


Figure 2: AME GUI After Matching

(ATE) and average treatment effect on the treated (ATT) are also indicated [k]. The GUI also provides the user with a number of useful tools for further analyzing the data and matching process. The “Matched Groups” tab [m] shows the matched group of a specified unit, along with its conditional ATE (CATE), where the unit is specified by its index [n]. A unit’s CATE is the expected treatment effect for that unit and others with identical covariates values. The “Graphs” tab creates visualizations to assess the importance of each covariate and the distribution of CATEs [o]. Finally, the “Matching Logs” tab [p] provides information pertaining to decisions the algorithm made in each iteration, including which covariate was dropped, the number of matches made, and the balancing factor.

#### 4. Conclusion

The AME-GUI provides a simple and ready-to-use overview of the usage and outputs of the open-source Python and R packages **dame-flame** and **FLAME** (Gupta, 2020; Orlandi et al., 2021). It does so on pre-loaded datasets, which users can experiment with in order to determine if **dame-flame** or **FLAME** would be suitable for their own data and applications. Both packages also include another almost exact matching algorithm called *DAME (Dynamic Almost Matching Exactly)* (Dieng et al., 2019), which is guaranteed to find the optimal subset of covariates to match on for each unit, whereas FLAME greedily approximates the optimal solution. Detailed tutorials are available at the AME website (AME-Lab). Future GUI extensions will support user-uploaded datasets and the DAME algorithm. We also plan to support our other almost exact matching algorithms, including Matching After Learning To Stretch (MALTS) and Adaptive Hyper-Boxes (AHB), which can handle mixed (categorical and continuous) covariates (Morucci et al., 2020; Parikh et al., 2018).

#### References

AME-Lab. <https://almost-matching-exactly.github.io/>.

- Stef van Buuren and Karin Groothuis-Oudshoorn. `mice`: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, pages 1–68, 2010.
- Awa Dieng, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable Almost-Exact Matching for Causal Inference. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2445–2453, 2019.
- Neha Gupta. `dame-flame` (0.4.1), 2020. URL <https://pypi.org/project/dame-flame/>.
- Marco Morucci, Vittorio Orlandi, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Adaptive Hyper-box Matching for Interpretable Individualized Treatment Effect Estimation. In *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020*, volume 124 of *Proceedings of Machine Learning Research*, pages 1089–1098. AUAI Press, 2020. URL <http://proceedings.mlr.press/v124/morucci20a.html>.
- Vittorio Orlandi, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. *FLAME: Interpretable Matching for Causal Inference*, 2021. URL <https://CRAN.R-project.org/package=FLAME>. R package version 2.1.0.
- Harsh Parikh, Cynthia Rudin, and Alexander Volfovsky. MALTS: Matching after learning to stretch. *arXiv preprint arXiv:1811.07415*, 2018.
- Paul R Rosenbaum and Donald B Rubin. The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1):41–55, 1983.
- Donald B Rubin. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *Journal of the American Statistical Association*, 100(469), 2005.
- Tianyu Wang, Marco Morucci, M. Usaid Awan, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. FLAME: A Fast Large-scale Almost Matching Exactly Approach to Causal Inference. *Journal of Machine Learning Research*, 22:31:1–31:41, 2021. URL <http://jmlr.org/papers/v22/19-853.html>.