Appendix B. Data Processing and Matching

Here we present the step-by-step methodology we undertook to match our drugs across global regulators. We also provide code snippets from the algorithm ue used to facilitate reproducibility and transparency of the matching process.

Before matching across regulatory databases, we first needed to check for duplicate entries in our data. Duplicates were most common in the data extracted from Global Data (Saudi, US and UK). While the data were relatively standardized in structure, it appeared that Global Data counted separate approvals for the same drug as a new entry when the drug was approved for a new therapy indication. Thus, to avoid over inflating our estimates, we had to ensure that each entity was uniquely presented across all datasets. Duplicates were defined as those having the same active ingredient, strength, dosage form and company. Products with any differences in any of these areas were regarded as distinct and hence retained in the analysis. Initially, the total number of branded and biologic medications retrieved from Global Data for the Saudi market for 2016-2024 was 1,668. After duplicates were removed, the number of unique entries was 1,167, which served as our base dataset subsequent matching.

Step 1: Standardized Identifiers

The first step of our matching process was done using unique identifiers. For the EMA data, we used EMA_Brand_ID, which was the only unique identifier available. A total of 645 drugs were identified as a perfect match between Saudi and EMA approval data. The following Python code illustrates the left merge that matched the EMA approval years:

```
# Perform a left merge to add EMA_Approval_Year to Saudi data based on EMA_Brand_ID
saudi_data = pd.merge(
    saudi_data,
    ema_data[["EMA_Brand_ID", "EMA_Approval_Year"]],
    on="EMA_Brand_ID",
    how="left",
)

#how many rows have been merged between saudi and EMA data
print(saudi_data["EMA_Approval_Year"].count())
```

```
645
```

For US FDA data, we performed two separate merges: one using the International_Drug_ID and another using the EMA_Brand_ID and created separate columns for each allowing us to cross-validate the two sources.

```
#left merge to add US Approval Year to the Saudi Data based on International_Drug_ID
saudi_data = pd.merge(
    saudi_data,
    uniqe_international_usfda_data[["International_Drug_ID", "US_Approval_Year"]],
    on="International_Drug_ID",
    how="left",
)

saudi_data = saudi_data.rename(columns={'US_Approval_Year': 'US_Approval_Year_International'})

#left merge to add US Approval Year to saudi_data based on EMA_Brand_ID
saudi_data = pd.merge(
    saudi_data,
    uniqe_EMAID_usfda_data[["EMA_Brand_ID", "US_Approval_Year"]],
    on="EMA_Brand_ID",
    how="left",
)

saudi_data = saudi_data.rename(columns={'US_Approval_Year': 'US_Approval_Year_EMA'})
```

These matches yielded 686 using EMA_Brand_ID and 802 matches using the International_Drug_ID. We used the same dual-matching approach with the UK data as well and got 915 matches using EMA_Brand_ID and 369 using International_Drug_ID. We then cross checked the years of approval we got for US and UK using the two different identifiers, and while most yielded the same result, in the cases of discrepancies we resorted to the official regulatory authority's website (US FDA or MEHRA) to verify the approval date.

Following all these steps, we still had over 270 drugs missing approval dates from all three regulators

```
#Count number of rows that don't have any of the specified approval years
missing_approval_years = saudi_data[
    saudi_data['EMA_Approval_Year'].isnull() &
    saudi_data['US_Approval_Year_International'].isnull() &
    saudi_data['US_Approval_Year_EMA'].isnull() &
    saudi_data['UK_Approval_Year_International'].isnull() &
    saudi_data['UK_Approval_Year_EMA'].isnull()
]

print("Number of rows without any approval years:", len(missing_approval_years)

Number of rows without any approval years: 276
```

Step 2: Fuzzy Matching

As we examined the datasets, we noticed a consistent pattern that drugs with the same brand name, applicant (company), and strength tended to be the same product in each of the regulators even in the absence of standardized IDs. This led us to use this for the fuzzy matching logic for the 276 drugs that left unmatched. We used fuzzy instead of exact matching here because brand

names, company names, and products strengths are not always recorded in the same way across regulatory datasets. In some instances, the same product may be marketed under slightly different names, or recorded with differences in formatting (e.g., "GlaxoSmithKline" vs. "GSK", or "Paracetamol 500mg" vs "Paracetamol 500 mg"). While these are minimal inconsistencies, still such small differences will break using an exact match algorithm. Further, we used 80% as our threshold and those with a matching score of 80% and above were mapped, while anything less was left empty. This process added an additional 52 matches reducing the number of unmatched records to 224. The code below demonstrates how fuzzy matching was performed for US data. The same methodology was applied across all relevant global datasets

```
#Create a new column to store the fuzzy matching results
saudi_data['fuzzy_matched'] = None

for index, saudi_row in saudi_data.iterrows():
    best_match = None
    highest_score = 0

    for _, us_row in usfda_data.iterrows():
        #combine fields for comparison
        saudi_combined = f"{saudi_row['Brand_Name']} {saudi_row['Strength']} {saudi_row['Applicant']}"
        us_combined = f"{us_row['Brand_Name']} {us_row['Strength']} {us_row['Applicant']}"

        #calculate the fuzzy match score
        score = fuzz.token_sort_ratio(saudi_combined, us_combined)

        # Update the best match if the score is higher
        if score > highest_score:
            highest_score = score
            best_match = us_row['US_Approval_Year']

    #Add the match to the new column if the score is above a threshold
    saudi_data.at[index, 'fuzzy_matched'] = best_match if highest_score > 80 else None
```

```
#Count the number of rows where fuzzy_matched is not null
fuzzy_matched_count = saudi_data['fuzzy_matched'].notnull().sum()

print("Number of rows fuzzy matched:", fuzzy_matched_count)

Number of rows fuzzy matched: 52
```

It is also important to mention that while this method accounts for slight differences in textual variation, it doesn't account for products which are marketed under completely different brand names in different countries (e.g. Vyvanse vs Elvanse), which were outside the scope of the automated matching and would have to undergo additional manual review.

Step 3: Manual Search

After fuzzy matching was conducted, we had a total of 224 unmatched drugs, so we reviewed these by conducting a manual search. The manual search involved looking through regulatory databases as well as searching companies' websites for general announcements about drug approvals. Manual inspection was a crucial step given that a subset of the drugs retrieved from Global Data were missing the standardized identifiers. While it is still not clear why some drugs did not have identifiers, in many instances, we were still able to find the products by searching the official source (US FDA, EMA, MEHRA) directly. Ultimately, we were able to recover approval year information provided by one or more of these three regulators for an additional 127 drugs.

After all the matching steps were completed, we were left with 57 drugs that were missing an approval date from either of these three regulators. This is mostly due to ambiguous records that could not be confirmed. For instance, some of the drugs sold under different brands in different countries, even had the formulation or dosage differ slightly. Hence it wasn't clear if they were the same product. Moreover, in some cases, the approval listed in one of the databases was for a variation (e.g. new indication or new formulation) rather than the original approval. The variation made it complicated to confidently gauge the exact approval date and benchmark it across regulators. Thus, we decided to exclude these drugs from the analysis. We also excluded an additional 40 drugs that did not have price information, primarily because they were withdrawn from the Saudi market and therefore, did not appear in the most recent price registries. Ultimately, the final analytic sample for the study consisted of 1,070 unique drugs.