CodeRx Medication Diversification Tool Tip Sheet

Create your own realistic pediatric asthma medication data by following this guide to combine the output of the CodeRx Medication Diversification Tool (MDT) with Synthea!

Find more information about this project at https://coderx.io/projects/medication-diversification. Thank you for your interest in our work!

Prerequisites

GitHub - https://github.com/signup

Java 1.8 or above

Windows PowerShell or Git Bash (or a Linux terminal)

Synthea Instructions

To clone the Synthea™ repo, then build and run the test suite:

```
git clone https://github.com/synthetichealth/synthea.git
cd synthea
./gradlew build check test
```

In Synthea, change setting in `synthea/src/main/resources/synthea.properties` to disable FHIR exporting and enable CSV exporting.

```
...
exporter.fhir.export = false
...
exporter.csv.export = true
...
```

- Each time you run Synthea, make sure you have all Synthea CSV output files closed, or it will error out with a non-specific error message.
- Run Synthea with a large enough sample size (at least 1000) to see a noticeable impact from MDT.
- Check the `medications.csv` output file for medications produced by your MDT-generated module.

```
./run_synthea -p 5000
```

NOTE: Change the number after -p to however big you want your population to be.

Check synthea/output/csv/medications.csv, filter column REASONDESCRIPTION = “Childhood asthma”, and note that there is only one product prescribed (Flovent 44mcg) for all patients with childhood asthma. You may want to save a copy of this file in a different directory to compare against the output of MDT + Synthea.

MDT Instructions

1. Clone the repo.

```
git clone https://github.com/coderxio/medication-diversification.git
cd medication-diversification
```

2. Create and activate a venv.

```
python -m venv venv
source venv/bin/activate
```

Or on Windows (using Git Bash):

```
py -m venv venv
venv/scripts/activate
```

If using VSCode on Windows and getting error "Activate.ps1 is not digitally signed. You cannot run this script on the current system.", then you may need to temporarily change the PowerShell execution policy to allow scripts to run. If this is the case, try:

`Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process` and repeat step 2.

3. Install MDT as an installed editable package (note the `.` after `-e`).

```
pip install -e .
```

4. Change to a new directory outside of the `medication-diversification/` project folder to test out MDT.

```
cd ..
mkdir mdt-test
cd mdt-test
```

5. Initialize MDT. This only needs to be done once. This will create a `data/` directory and load the `MDT.db` database. **This takes a few minutes to complete.**

```
mdt init
```

6. Create a new module. This will create a `<<module_name>>/` directory which is empty except for an initial `settings.yaml` file.

```
mdt module -n maintenance_inhaler create
```

7. Edit the `settings.yaml` folder in the newly created `<<module_name>>/` directory, following the directions in this README.
    a) For this example, just copy and paste the file from our GitHub docs/examples folder: https://raw.githubusercontent.com/coderxio/medication-diversification/main/docs/examples/maintenance_inhaler/settings.yaml
    b) Review other MDT settings in our README: https://github.com/coderxio/medication-diversification#user-defined-settings
8. Build the module.

```
mdt module -n maintenance_inhaler build
```

This will create:

- A `maintenance_inhaler.json` file which is the Synthea module itself
- A `lookup_tables/` directory with all transition table CSVs
- A `log/` directory with helpful output logs and debugging CSVs

Repeat steps 7 and 8 until MDT is producing medications that align with what you would expect. Use the `log <<timestamp>>.txt` files in the `log/` directory as a quick and easy way to validate the output of the module with a clinical subject matter expert.

To create a new module, start at step 6.

Integrating MDT Module Output with Synthea

Using the existing asthma module as an example...

1. Open the file `synthea/src/main/resources/modules/asthma.json`
2. Change this...

```
...
    "Prescribe_Maintenance_Inhaler": {
      "type": "MedicationOrder",
      "reason": "asthma_condition",
      "codes": [
        {
          "system": "RxNorm",
          "code": "895994",
          "display": "120 ACTUAT Fluticasone propionate 0.044 MG/ACTUAT Metered Dose
Inhaler"
```

```
        }
      ],
      "prescription": {
        "as_needed": true
      },
      "direct_transition": "Prescribe_Emergency_Inhaler",
      "chronic": true
    },
...
```

To this...

```
...
    "Prescribe_Maintenance_Inhaler": {
      "type": "CallSubmodule",
      "submodule": "medications/maintenance_inhaler",
      "direct_transition": "Prescribe_Emergency_Inhaler"
    },
...
```

3. And make sure your submodule JSON and transition table CSVs are in the folder locations specified above.

   a) Copy-paste the `mdt-test/maintenance_inhaler/maintenance_inhaler.json` file into the `synthea/src/main/resources/modules/medication` folder.

   b) Copy-paste all the transition table CSV files from `mdt-test/maintenance_inhaler/lookup_tables` into the `synthea/src/main/resources/modules/lookup_tables` folder.

See below for end result of example Synthea file structure:

```
synthea/
├─ src/
│  ├─ main/
│  │  ├─ resources/
│  │  │  ├─ modules/
│  │  │  │  ├─ medication/
│  │  │  │  │  ├─ maintenance_inhaler.json
│  │  │  │  │  ├─ ...
│  │  │  │  ├─ lookup_tables/
│  │  │  │  │  ├─ maintenance_inhaler_ingredient_distribution.csv
│  │  │  │  │  ├─ maintenance_inhaler_fluticasone_product_distribution.csv
│  │  │  │  │  ├─ maintenance_inhaler_budesonide_product_distribution.csv
│  │  │  │  │  ├─ maintenance_inhaler_beclomethasone_product_distribution.csv
│  │  │  │  │  ├─ ...
│  │  │  │  ├─ asthma.json
│  │  │  │  ├─ ...
```

4. Lastly, if the calling module (in this case, `asthma.json`) ends medications by a specific `State_Name` of a previous `MedicationOrder` state, you will need to change that `MedicationEnd` state to instead end a medication by `attribute`. The reason for

this is that our MDT JSON module generates different `MedicationOrder` state names for each potential prescribed product, but they all have the same `attribute`.

a. Open `synthea/src/main/resources/modules/asthma.json` again and change this…

```
...
    "Maintenance_Medication_End": {
      "type": "MedicationEnd",
      "medication_order": "Prescribe_Maintenance_Inhaler",
      "direct_transition": "Emergency_Medication_End"
    },
...
```

To this…

```
...
    "Maintenance_Medication_End": {
      "type": "MedicationEnd",
      "referenced_by_attribute": "maintenance_inhaler",
      "direct_transition": "Emergency_Medication_End"
    },
...
```

Now you have updated the Asthma module in Synthea with the outputs of the MDT module. If you re-run Synthea like you did at the beginning, you should expect to get different mediation outputs.

Go back to your Synthea directory in PowerShell or Git Bash or Linux terminal, and do the following.

```
./run_synthea -p 5000
```

`NOTE: Change the number after -p to however big you want your population to be.`

NOTE: Make sure you closed the medications.csv file from the first time you ran Synthea, otherwise you might get errors relating to CSVExporter and a failed build overall.

Check `synthea/output/csv/medications.csv` , filter column REASONDESCRIPTION = “Childhood asthma”, and note that there is now a variety of maintenance inhaler medication product outputs in the DESCRIPTION column. The CODE column contains the RxNorm RXCUI for these products.

If you open your `mdt-test/maintenance_inhaler/log/log <<timestamp>>.txt` file, at the bottom, you should see a breakdown of the different distributions of products you should expect to see in your Synthea output. Note, however, that Synthea adds a layer of randomness… so depending on your population size, you might not see all of them, or this exact distribution breakdown. But this should give you an idea of the types of medication products to look for in your medications.csv output file.

Thanks again for following along and find more information about CodeRx at https://coderx.io/

Get more detailed developer info at our open source GitHub repo for this project:

https://github.com/coderxio/medication-diversification