

Creating Synthetic Data that Mimics Real World Data for Accurate Analysis Programming Development

Anti-drug Antibody (ADA) Data as a Case Study

Main Author: Linshan Yuan (LYuan7@ITS.JNJ.com), Co Author: Weiwei Xu (wxu52@ITS.JNJ.com)

Background & Objectives

Development of analysis programs often requires realistic test data before final study data becomes available. This poster presents effective methods for generating synthetic ADA data.

- ADA Analysis Complexity: Multiple correlated test outcomes including both numeric and character results
- Challenge: Need for realistic synthetic data that preserves internal logical relationships
- Objective: Create reliable analysis datasets that accelerate TFL generation

Methods & Approach

Algorithm Development:

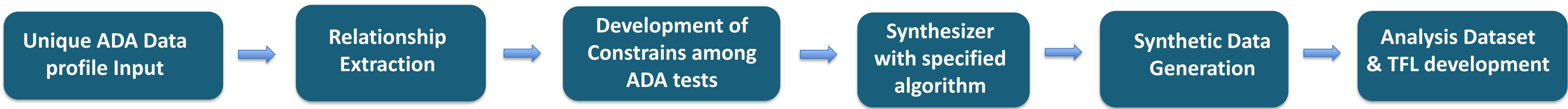
- Utilize real-world ADA data as foundation: The all-related tests from ADA profile are converted into tabulation structure within a subject. The unique tabulations are identified based on shape meaning across the ADA outcomes and the numbers of records for visits.
- Preserve internal logical and complex relationships: combine two types of information: metadata extracted by a specific SDV agent function, and general knowledge about the ADA tests (for example, dependencies and value constraints
- Modify to align with specific study design: map the synthesized data, preserving its shape, to each subject according to their corresponding planned visits.
- Customize for investigated drug characteristics: synthesizer with customized constrains can provide any shape of tabulation ADA profile according to the study design.
- Adapt to specific analysis scheme requirements

Key Benefits

Key Advantages of Synthetic Data Approach:

- Accelerated Development: Foster the development of analysis outputs (TFL) without real data
- Minimal Code Changes: Analysis programs require few modifications when final data arrive and shorten the ADA analysis delivery within a week
- Flexibility: fully Adaptable to various study designs with different schedule and various data cut and drug characteristics

Data Flow & Implementation



```
# Create and train the synthesizer with constraints
# Note: Order matters:|
# 1. constraint1 (ADATTR related to ADACNCLS): fills ADATTR for Positive rows of ADACNCLS
# 2. constraint2 (SPTITER to ADTTR): forward-filland only updates when ADATTR has a value LARGER than current SPTITER
# 3. constraint3 (SUADAST to SPTITER): sets SUADAST based on SPTITER
# 4. constraint4 (ADTTR to ADACNCLS): sets ADATTR=NaN for Negative rows
# 5. constraint5 (NAB to ADACNCLS): sets NAB based on ADACNCLS
# 6. constraint6 (allowed values to ADATTR): ensures ADATTR is one of allowed values (10, 20, 40, 80, 160)
# 7. constraint7 (SUNABST to NAB): ensures SUNABST equals NAB after NAB is set
```

Tabulation data for dummy_001

Original dataframe shape: (12, 8)							
	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST seq
0	dummy_001	Negative	NaN	NaN	NaN	NaN	NaN 1
1	dummy_001	Negative	NaN	NaN	Negative	NaN	NaN 2
2	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative 3
3	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative 4
4	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative 5
5	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 6
6	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 7
7	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 8
8	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 9
9	dummy_001	Positive	10.0	10.0	Positive	Positive	Positive 10
10	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 11
11	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN 12

Tabulation data for dummy_002

Original dataframe shape: (10, 8)							
	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST seq
0	dummy_002	Negative	NaN	NaN	NaN	NaN	NaN 1
1	dummy_002	Positive	20.0	20.0	Positive	Negative	Negative 2
2	dummy_002	Positive	10.0	20.0	Positive	Positive	Positive 3
3	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 4
4	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 5
5	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 6
6	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 7
7	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 8
8	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 9
9	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN 10

Synthetic data for dummy_001

Synthetic dataframe shape: (15, 8)										
Generated 15 rows of synthetic data:										
	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST	seq		
0	dummy_001	Negative	NaN	NaN	Negative	NaN	NaN	1		
1	dummy_001	Negative	NaN	NaN	Negative	NaN	NaN	2		
2	dummy_001	Negative	NaN	NaN	Negative	NaN	NaN	3		
3	dummy_001	Negative	NaN	NaN	Negative	NaN	NaN	4		
4	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative	5		
5	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN	6		
6	dummy_001	Positive	10.0	10.0	Positive	Positive	Positive	7		
7	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN	8		
8	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative	9		
9	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN	10		
10	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative	11		
11	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN	12		
12	dummy_001	Positive	10.0	10.0	Positive	Positive	Positive	13		
13	dummy_001	Negative	NaN	10.0	Positive	NaN	NaN	14		
14	dummy_001	Positive	10.0	10.0	Positive	Negative	Negative	15		

Synthetic data for dummy_002

Synthetic dataframe shape: (17, 8)										
Generated 17 rows of synthetic data:										
	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST	seq		
0	dummy_002	Positive	10.0	10.0	Positive	Negative	Negative	1		
1	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	2		
2	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	3		
3	dummy_002	Positive	20.0	20.0	Positive	Negative	Negative	4		
4	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	5		
5	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	6		
6	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	7		
7	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	8		
8	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	9		
9	dummy_002	Positive	10.0	20.0	Positive	Positive	Positive	10		
10	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	11		
11	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	12		
12	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	13		
13	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	14		
14	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	15		
15	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	16		
16	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	17		

Note: One original tabulation data could create any number of tabulation with different shape and multiple unique original ADA tabulations could synthesize even more different tabulation data and provide the most various dummy ADA profiles.

Employ Python's SDV to build and train the synthesizer with constraints; the ADA features and domain knowledge are maintained during the synthetic process and dynamically adjusted to each subject's time points based on progress or protocol.

Realistic Data

Synthetic data maintains statistical properties and logical relationships of real ADA data

Efficiency Gain

Significantly accelerates analysis program development and testing cycles

Seamless Transition

Minimal adjustments needed when switching from synthetic to actual study data