Paper PP07

Optimizing Flexibility and Consistency in Statistical Programming for Platform Trials: Practical Approaches to Data Cut, SDTM, ADaM, and TLF Setup

Zhen (Laura) Li, AstraZeneca, Gaithersburg, USA
Kelly Fang, AstraZeneca, Wilmington, USA

ABSTRACT

Platform trials, a type of master protocol trial, provide complexity and adaptability in drug development by supporting multiple sub-studies under a unified infrastructure with sub-study-specific protocols. Though this enhances efficiency and innovation, it poses distinct challenges for statistical programming including dynamic analysis requirements, competing timelines, and adapting to evolving protocols. This poster shares practical, scalable solutions from an oncology platform trial, designed to optimize efficiency, flexibility, and consistency across the data lifecycle. Adaptive strategies for raw data cuts allow rapid, precise data selection for varied deliverables, alongside SDTM specification and programming to harmonize data structures across sub-studies. Our ADaM specification and programming provide a consistent, yet flexible framework across sub-studies for various requirements and separate CSRs; with TLF development streamlining output generation accordingly. Examples from our study demonstrate how these strategies improve resource allocation, timeline control, and process efficiency, offering practical takeaways for programming teams working in complex, adaptive trials.

INTRODUCTION

Master protocol trials have emerged as an innovative strategy in drug development that enable sponsors to evaluate multiple therapeutic hypotheses within a unified operational and statistical framework. By coordinating design, infrastructure, governance, and analyses across sub-studies, these trials accelerate evidence generation while preserving scientific rigor. Although these features drive efficiency, platform trials also create unique challenges for statistical programming. Teams must manage heterogeneous data structures, accommodate frequent protocol and CRF amendments, and deliver overlapping analyses on compressed timelines. This paper focuses on practical, scalable programming solutions for platform trials with similar characteristics, illustrating how thoughtfully designed data cuts, unified SDTM, and sub-study–specific ADaM specifications, programming and TLF programs set-up can preserve flexibility without sacrificing consistency. Drawing on experience from an in-house oncology platform trial, we describe approaches that shorten cycle times, improve traceability, and enable parallelized delivery across evolving sub-studies according to the characteristics of data and analyses requirements.

MASTER PROTOCOL TRIALS

Master protocol trials are increasingly used to accelerate drug development by evaluating multiple therapies and/or disease subpopulations within a single overarching protocol. These designs enable coordinated evaluation of one or more interventions within a single clinical trial framework across one or more patient populations that share common enrollment, follow-up, and assessment procedures. Under a unified governance structure, multiple sub-studies can be integrated, each pursuing distinct objectives while leveraging shared control arms, common protocol elements, centralized infrastructure, and consistent oversight. This design facilitates expedited drug development, reduces participant burden, enhances cost-effectiveness, and operational efficiencies. Conversely, complexity can arise from evolving sub-studies, heterogeneous analysis definitions, frequent protocol and CRF amendments, and overlapping timelines across sub-studies.

Examples of master protocol trial types include umbrella, basket, and platform trials:

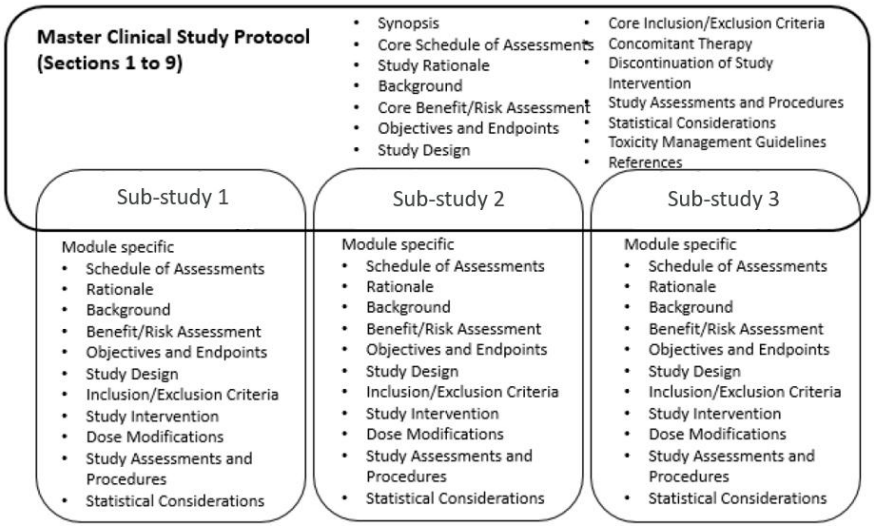
- Basket trial: a trial designed to evaluate a medical product for multiple diseases, conditions, or disease subtypes.
- Umbrella trial: a trial designed to evaluate multiple medical products concurrently for a single disease or condition.
- Platform trial: a trial designed to evaluate multiple medical products for a disease or condition in an ongoing manner, with medical products entering or leaving the platform.

A PLATFORM TRIAL

This is an oncology phase II platform trial, and all programming work is conducted by our internal programming team. Two sub-studies were initiated in 2023 and a third was added in 2024, reflecting the adaptive nature of this platform trial. Sub-study 1 evaluates monotherapy across multiple dose levels; Sub-study 2 investigates combination regimens with different chemotherapy agents; and Sub-study 3 evaluates monotherapy across multiple parts. Each sub-study targets a distinct indication. This study comprises four protocols: one master protocol and three sub-study-specific

protocols. The master protocol defines the overarching trial framework, including general study conduct and governance, as well as design elements common across all sub-studies. Each sub-study is further governed by its own clinical study protocol, which sub-study-specific details such as indication-specific eligibility criteria, treatment regimens, dosing schedules, and clinical assessment schedules. While the master protocol establishes consistency across the study, the sub-study protocols allow for tailored designs aligned with distinct scientific objectives. Figure 1 below outlines the four protocols.

Figure 1: Protocol Structure



CHALLENGES

Heterogeneous data poses a challenge for this trial. This study uses a single raw database to capture data across all sub-studies. The CRFs include forms that apply across all sub-studies as well as forms and individual fields that are specific to a given sub-study (For example, eligibility criteria, exposure, and adverse events). Consequently, some data-mapping rules differ across sub-studies.

Due to the heterogeneity across sub-studies and treatment arms, some analysis definitions vary and are more complex. For example, patients were either randomized or assigned into arms, which required careful handling of certain analysis set definitions, such as full analysis set, to account for the differences between these scenarios. Each sub-study has a distinct study schema, resulting in sub-study-specific visit window definitions driven by differing schedules and dynamic rules, including cycle lengths, assessment frequencies, and evaluation periods. In addition, because the dosing schedules and combination regimens vary across chemotherapy agents, the definition of the DLT evaluable set is complex, as it varied not only across sub-studies but also across arms within a given sub-study. Arm-specific criteria had to be carefully applied with respect to minimum safety evaluation periods, cycle length, dosing requirements, programmatic derivation of complete infusions.

As new sub-studies may be added and existing sub-studies amended over the course of the study, this study experienced more frequent data migrations than a traditional single-study design. Due to frequent data migrations and high analysis demand, which will be detailed later, the timing of implementing programming updates without impacting analysis requests is critical. Appropriate timing is determined by carefully assessing the analysis timelines for all sub-studies and evaluating downstream impacts across them—including updates to the annotated CRF, mapping specifications, datasets, and TLFs.

Additionally, differences in analysis requirements across sub-studies—including milestone-triggered analyses and ad hoc analyses for multiple objectives—introduce additional complexity. With the exception of a few pooled analyses (e.g., DSUR), most analyses require results from specific sub-studies or arms. Planned analyses include regular SRC reviews, sub-study-specific interim analyses, arm-specific interim analyses, and sub-study-level CSRs. Multiple interim analyses conducted for each sub-study or arm prior to a sub-study's final CSR. For these interim analyses, Data cut-off requirements vary: some require selecting data from one or two sub-studies, others by specific arms, others strictly by a cut-off date, and some by a combination of these criteria. Subgroup analyses also vary across sub-studies due to differences in patient populations, disease characteristics, and treatment history. For example, primary tumor locations of interest differ by sub-study, resulting in subgroup variables that are defined and analyzed only where relevant.

In addition to the requested analyses, dry-run planning, including timeline and approach, is also comparatively challenging because it must balance programming efficiency and dry-run review effectiveness while accounting for all sub-studies, even though TLFs are separated by sub-study. Some TLFs—either entire outputs or specific rows—are common across sub-studies or arms, while others are unique to a given sub-study or arms. For dry-run planning, we conduct a full dry-run of all TLFs for the earliest interim analysis in the study. For interim analyses in a new sub-study, targeted dry-runs are performed for selected outputs, chosen based on:

- Differences from the initial dry-run: extent of divergence in TLF structure or content across sub-studies;
- Programming complexity and risk: potential for logic errors or intricate derivations stemming from differing definitions across sub-studies;
- Interpretive significance: key outputs that materially impact decision-making

Numerous ad hoc requests with diverse specifications are generated, often with competing timelines across sub-studies. The programming team frequently processes multiple deliverables in parallel, necessitating context-switching between sub-studies. As analyses may target any sub-study, the request volume is higher than in a single-study trial. For supporting ad hoc analyses, an important question throughout the trial is how to utilize limited resources to process deliverables as efficiently as possible without delaying any timelines—and in turn decision-making. To balance these demands, the technical approach should evaluate which components can be synchronized across sub-studies and which must remain separate when setting up programs.

TECHNICAL SOLUTIONS

After raw data are received, the standard analysis workflow proceeds as follows: apply data cuts (as needed), prepare SDTM datasets, prepare ADaM datasets, and generate TLFs. This workflow remains unchanged; however, the platform trial's data structures and deliverable requirements necessitate specific considerations and targeted technical approaches to enhance programming efficiency. This section outlines the technical approaches we implemented.

DATA-CUT APPROACHE

Cutting raw data focuses on the clean, relevant records needed to ensure analytical reliability and interpretability. We aim to design a single approach that covers all data-cut scenarios, maximizing adaptability while minimizing complexity. Designing adaptive data-cut programs is essential to efficiently support various data-cutting requirements. The approach accounts for the following conditions:

- A single EDC database spanning all sub-studies;
- Analyses are at the sub-study or arm level;
- Data snapshots can be shared across multiple sub-studies or be sub-study specific

As an effective solution to support analyses with varying data selection requirements, two separate data-cut programs are created and validated separately:

1. Patient-level cut: selects patients by sub-study or arm across all raw datasets.
2. Observation-level cut: applies observation-level filters aligned to company-standard data-cut-off specifications, customized to accommodate study-specific deviations from company-standard raw data structure.

These separate programs enable programmers to efficiently select data by using either patient-level cut or observation-level cut or both, and to choose the sequence in which they are applied. Examples of applicable scenarios include:

- Analyses sharing a common DCO date and requiring reports for all sub-studies without patient selection: run the observation-level cut only.
- Analyses that do not require a DCO date but require subsetting by specific arms or sub-studies: run the patient-level cut only.
- Interim analyses for a particular sub-study with a specific DCO date: run both patient-level and observation-level cuts to satisfy criteria at both levels.

Additionally, the patient-level data-cut program is designed for flexibility across source datasets, beyond only cutting data at raw data level, allowing patient selection to be tailored to efficiently align with analysis requirements. For example, if an ad hoc analysis targets a specific patient group after an interim readout, the patient-level data-cut program can be applied to the interim ADaM datasets to shorten the programming timeline.

Traceability and efficient quality control are supported via a structured folder setup separating original raw data, post-cut datasets, and patient selection lists for each raw dataset.

SDTM DATA SPECIFICATION AND PROGRAMS

Because most SDTM mapping rules are shared across sub-studies, while certain rules or SDTM values are unique to specific sub-studies—and because CRF migrations are frequent in platform trials and drive additional data-mapping amendments— we implemented a unified SDTM specification and a single program set maintained centrally to streamline maintenance, ensure consistency, and support harmonized updates across sub-studies, maximizing efficiency in programming resource utilization, setup, updates, and quality control. These are augmented with variables capturing sub-study or arm information where relevant (e.g., TI, TV, and selected other domains).

ADAM DATA SPECIFICATION AND PROGRAMS

ADaM data-generation efficiency is the most critical driver of overall programming efficiency because analysis definitions are specified at the ADaM level. When designing ADaM programs setup and maintenance approach, two options were evaluated:

1. Single program set: one set of ADaM programs containing analysis definitions and variables for all sub-studies.
2. Separate program sets: distinct ADaM programs for each sub-study.

To support the decision, here are some considerations:

- A common SDTM structure across sub-studies.
- Most variables are consistent while accommodating sub-study and arm-specific analysis definitions.
- CSRs are produced per sub-study.
- Interim analysis reports and ad hoc may be arm-specific, sub-study-specific or data-pooling.

Our approach for data specification uses a single ADaM data specification file with separate derivation columns per sub-study (e.g., substudy_1, substudy_2, substudy_3). Cells remain blank for variables not applicable to a given sub-study. To leverage established macros (e.g., those that assign variable attributes based on the data specification) that assume a single derivation column in company-standard data specifications, we made minor updates to adapt these macros effectively to the multi-column structure.

Given the separate CSRs, high analysis demand, and the addition of new sub-studies, we selected Option 2 to avoid overly complex, large SAS programs that would extend setup and maintenance timelines. Under Option 2, ADaM programs are created and maintained per sub-study to reflect distinct analysis definitions and to minimize the complexity of each SAS program. This approach allows programmers to focus only on programming updates needed for the specific sub-study associated with each analysis request. A separate set of ADaM programs for each sub-study also allows the study programming team to process requests in parallel across sub-studies, accelerating deliverables with competing or overlapping timelines.

ADaM programs are managed within sub-study–specific directories, unlike SDTM specifications and programs, which are maintained in a central location across all sub-studies. This structure supports independent analysis development and is aligned with the preparation of separate clinical study reports.

TLF PROGRAMS

Because treatments and certain TLFs (or specific rows within TLFs) differ by sub-study, TLF programs were also created per sub-study, based on considerations similar to those used in the ADaM program setup approach. This approach also enables efficient reuse of department-developed template programs.

TLF programs are centrally stored for each sub-study to sustain consistency and ease of maintenance.

ADVANTAGES

The chosen technical approaches reduce redundant updates and improve traceability across evolving sub-studies. A unified data-cut and SDTM approach minimizes effort on program maintenance and lowers the risk of inconsistent updates and quality control. Two complementary data-cut programs support diverse data-cut requirements and allow flexible selection of source datasets, reducing development effort and rework. Per-sub-study ADaM and TLF programs help contain complexity as the platform expands, while reuse of templates accelerates setup.

We maintain central program repositories. With SDTM programs stored in a single shared repository regardless of sub-study, while ADaM and TLF programs are stored in central repositories at sub-study level. For each deliverable, programs are copied from the central folder, executed, and—post-delivery—reconciled back to central storage by the study lead programmer to ensure the latest programming logic is retained. Each deliverable follows this standard process from cutting raw data (if needed) to TLF generations to maximize traceability.

To support overlapping timelines across sub-studies, data-cut and SDTM preparation typically involve one primary and one validation programmer, with an additional programmer added as needed for parallel work on ADaM and

TLFs generations. For ADaM and TLFs, we commonly assign one primary programmer and two validators, reflecting the higher validation effort. This structure enables parallelized delivery across sub-studies while accelerating overlapping timelines and enhancing programming resource utilization efficiency.

CONCLUSION

With a team of two to three study programmers and one study lead programmer, we delivered 35+ high-quality analyses in 2025, achieving turnarounds as short as 1.5 days while managing numerous overlapping and competing timelines for this in-house study. Thoughtful technical planning—including adaptive data cuts, unified SDTM, sub-study-specific ADaM specifications and program organization, and centralized governance—significantly improved quality and efficiency.

We leveraged technical approaches to identify the most efficient analysis paths, enabling the study team to meet all decision-making objectives—and, at times, to deliver requests with timelines that initially seemed unrealistic—without significantly increasing programming resources. For example, when analyses spanned multiple sub-studies with rapid turnaround and DCO dates that were identical or close to one another, data-cut and SDTM programs spanning all required sub-studies were pre-executed in centralized folders to minimize discrepancies between primary side and validation side ahead of the official snapshot. At the official snapshot date, these programs were copied and re-run rapidly in the specific sub-study request folder to ensure adherence to each request's standard data flow. ADaM and TLF programming for each sub-study proceeded in parallel. This approach ensured traceability, reduced timeline risk, and limited upfront preparation, avoiding the intensive programming effort typically required.

From a resourcing perspective, deliverables spanning multiple sub-studies with competing timelines typically require two programmers for data cuts and SDTM preparation—one on primary side and one from validation side. For ADaM and TLFs, we assign one primary programmer and two validation programmers, given validation is usually more time-consuming. In 2025, we delivered seven interim analyses and, for each, successfully handled at least two ad hoc requests in parallel. Our chosen technical approach directly shapes resource allocation, timelines, and how we manage deliverables with competing or overlapping schedules.

Given the high demand for analysis deliverables with overlapping timelines, effective communication on priorities and the objective of each deliverable is both challenging and critical. To ensure assignments are clear to all study programmers, we use a Planner in Teams to track resource allocation and progress for each analysis deliverable. We maintain two Planner tabs: a study-level view and a project-level view. The study-level tab organizes requests into Preparation, In Progress, Reviewing, and Completed. The project-level tab lists all planned deliverables for the year, giving our study programming team a forward view of workload for planning time-off and helping the Global Product Programmer and Study Lead Programmer track high-level status—such as whether all necessary materials are available to start preparation—and assess risk, for example, when a deliverable is the first for a new sub-study and therefore requires more preparation time and more thorough reviews and spot checks.

Comprehensive planning is critical for a platform trial, especially in the early phase, when initial programming setup is typically the most time-consuming. Beyond technical approaches, decisions on the programming preparation plan, such as dry-run timelines, and clear role assignments aligned to each programmer's strengths are also essential, as they form the foundation for an efficient study programming team.

As a platform trial may evolve to include additional sub-studies, early technical decisions may need to be re-evaluated and adjusted accordingly. Additionally, platform trials vary widely—for example, some use separate databases per sub-study, while others share a single database—so there is no one-size-fits-all solution. The key is to ask the right questions early, gather sufficient context, and tailor the technical approach, operational plan, and programming resource allocation to the dynamics of the specific trial.

REFERENCES

1. US Food and Drug Administration (FDA). Master Protocols for Drug and Biological Product Development: Guidance for Industry. Silver Spring, MD: FDA; 2023.
2. George Washington University School of Medicine & Health Sciences. Overview of Modern Clinical Trial Designs. Washington, DC: GW SMHS; 2023.
3. US Food and Drug Administration (FDA). Master Protocols: Efficient Clinical Trial Design Strategies to Expedite Development of Oncology Drugs and Biologics: Guidance for Industry. Silver Spring, MD: FDA; 2022.
4. Roustit M, Demarq O, et al. Platform trials. Therapies. 2023;78:29-38.

ACKNOWLEDGMENTS

We extend our sincere appreciation to all study programmers for their outstanding collaboration and dedication over the past two years. We also thank the Early Oncology Programming leaders for their support and trust, and for championing our continued growth.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Zhen (Laura) Li
Company: AstraZeneca
Email: zhen.li@astrazeneca.com

Co-Author Name: Kelly Fang
Company: AstraZeneca
Email: kelly.fang1@astrazeneca.com