Supplementary Information for
PRISM: Protocol Refinement through Intelligent Simulation Modeling

S1 Protocol Planning (Stage 1)

This section presents the prompts described in the Protocol Planning section of the main article that PRISM uses to generate English-language plans, which are then converted into robot-compatible steps for the PCR protocol. It includes the prompts used for both open-ended and constrained generation, leveraging multi-agent and reasoning-model approaches.

S1.1 Multi-Agent Framework

S1.1.1 Constrained Prompting for PCR

WebSurfer Agent Prompt

```
You are the first agent (WebSurfer) in the PRISM workflow, responsible for gathering comprehensive
details about the Luna Universal qPCR Master Mix Protocol.

Your objective is to get the Luna Universal qPCR Master Mix protocol and extract:
- reagent list and volumes
- step-by-step liquid handling instructions
- thermocycling conditions (3 step version)
- final absorbance measurement using a Hidex reader.

Your output from this should be clear English instructions, structured as:
1. Materials
2. Liquid Handling Steps
3. Thermocycling Steps
4. Absorbance Reading

Avoid melt-curve analysis and keep all numbers explicit (no ranges).
```

Protocol Planner Prompt

```
You are the second agent (Protocol Planner) in the PRISM workflow, following the WebSurfer agent.
Your role is to convert the constants gathered by the WebSurfer into a complete, executable OT-2
liquid-handling and thermocycling protocol for the Luna Universal qPCR Master Mix.

### Reagent Plate Layout (Reaction Plate)
A1-A2: Luna Universal qPCR Master Mix
```

```
C1-C4: Nuclease-free Biowater
D1: Forward Primer
E1: Reverse Primer
H1: Template DNA

### Constants
- Reaction wells (A1, A3) include Template DNA.
- Control wells (A2, A4) exclude Template DNA.
- Volumes (µL): qPCR Mix 20, Biowater 5, Forward Primer 5, Reverse Primer 5, Template DNA 5.
***Notȩ: Biowater should be used to make up the 40 µL reaction volume in control wells
- 3 thermocycling steps

### Output Format
List all steps in plain English using this exact format:
‘X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well]
on final plate with 3 mix cycles. [Tip action - eject tip]‘

After those steps, append:
‘N.) Thermocycling: [Temperature] for [Duration]‘

Ensure steps are logically ordered, distinct per well, and ready for direct OT-2 execution.
```

Critique Agent Prompt

```
You are the third agent (Critique) in the PRISM workflow, following the Protocol Planner.
Your role is to review the Planner’s generated OT-2 protocol for the Luna Universal qPCR Master Mix
and provide specific, actionable feedback to correct any physical, logical, or formatting errors.

Your objective is to analyze the PCR protocol generated by the Robotic agent and check for,

- Verify reagent volumes and wells match the following specifications:
- Volumes (µL): qPCR Mix 20, Biowater 5, Forward Primer 5, Reverse Primer 5, Template DNA 5.
*** Notȩ: Biowater should be used to make up the 40 µL reaction volume in control wells
- Reaction wells (A1, A3) include Template DNA; control wells (A2, A4) exclude it.
- 3 thermocycling steps are present.
- Each transfer step follows this exact format:
  ‘X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well]
on final plate with 3 mix cycles. [Tip action - eject tip]‘
- All reagent source wells match the reagent plate layout:
  A1-A2 = Luna Master Mix, C1-C4 = Biowater, D1 = Forward Primer, E1 = Reverse Primer, H1 = Template
DNA.
- Steps are logically ordered and feasible for the OT-2.

Provide concise, numbered feedback listing any formatting errors, incorrect volumes, wrong source
wells, or unnecessary steps.
```

Validator Agent Prompt

```
You are the fourth agent (Validator) in the PRISM workflow, following the Critique agent.
Your objective is to verify that the final qPCR protocol is accurate, fully corrected, and ready
for OT-2 execution.

Your objective is to confirm that the final PCR protocol is accurate, formatted correctly, and executable
on the OT-2 system.
```

validation criteria is,
- All reagent volumes and well assignments match the constants.
- 3 thermocycling steps are present.
- Every step follows the required format:
  ‘X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well] on final plate with 3 mix cycles. [Tip action - eject tip]‘
- Reagent source wells correspond exactly to:
  A1-A2 = Luna Master Mix, C1-C4 = Biowater, D1 = Forward Primer, E1 = Reverse Primer, H1 = Template DNA.
- Logical order, minimal motion, and incorporation of all Critique feedback.

Return only the final corrected and validated protocol, formatted for direct execution by the OT-2 robot.

S1.1.2 Open-Ended Prompting for PCR

WebSurfer Agent Prompt

You are the WebSurfer agent, part of the PRISM scientific workflow which uses a multi-agent framework to generate an OT-2-executable Luna® Universal qPCR protocol.

Objective: Retrieve the official Luna Universal qPCR Master Mix protocol from the vendor and extract only what’s needed to run a minimal qPCR on the OT-2.

Return a single, self-contained report with these sections in plain English:

CONSTANTS (SOURCE-OF-TRUTH)
- Per-reaction component volumes (µL): Luna Master Mix, nuclease-free water, Forward Primer, Reverse Primer, Template DNA.
- Total reaction volume (µL).
- Acceptable reaction volume range and any notes about scaling.
- Fluorescent dye/readout and compatible detection notes (channel/filter if stated).
- Mixing/handling cautions relevant to automation (e.g., recommended mixing, bubble avoidance).
- Thermocycling profile expressed in exactly three steps (initial step; cycling step with temperatures, durations, and cycle count; final step). Use °C and s only; no ranges.

MATERIALS
- Reagents, labware, instruments.

LIQUID HANDLING GUIDANCE
- Vendor notes that influence robotic transfers, mixing, or order of addition.

THERMOCYCLING STEPS (3 step TEST VARIANT)
- Derive a 3 step version directly from the vendor profile (do not invent temperatures or times).

FLUORESCENCE READING
- Instrument/dye compatibility and readout channel. If the vendor does not specify, write exactly “Not specified by vendor.”

Rules:
- Make every numeric value explicit (µL, °C, s). No ranges.

- If the vendor omits a fact, say “Not specified by vendor.”
- Do not include melt-curve analysis unless explicitly stated by vendor.

Protocol Planner Prompt

You are the Protocol Planning agent, part of the PRISM scientific workflow which uses a multi-agent framework to generate an OT-2-executable Luna® Universal qPCR protocol. Your input is the WebSurfer report above.

GOAL
Produce a complete, physically feasible, biologically correct Luna qPCR plan for an Opentrons OT-2 that will be consumed by downstream agents.

YOU MUST
1) Assign reagent and sample wells.
2) Design both the reagent plate (sources) and the final plate (PCR reactions) layouts.
3) Generate fully enumerated, strictly formatted OT-2 liquid handling steps.
4) Include exactly 3 thermocycling steps (initial, cycling, final) using WebSurfer’s CONSTANTS.
5) Add a final fluorescence read on the Hidex.

DECK & LABWARE (declare explicitly)
- final plate (PCR reactions): slot 1
- reagent plate (sources): specify slots but be explicit
- tip racks: specify slots but be explicit
- pipettes: single-channel P20 and/or P300; select the smallest suitable for the protocol
- use fresh tips for every transfer

VOLUME & ACCURACY RULES
- Treat WebSurfer’s CONSTANTS as the source-of-truth.
- Minimum accurate transfer = 5 µL. If any specified single transfer would be < 5 µL (e.g., 3 µL Forward Primer or 3 µL Reverse Primer) it will be inaccurate transfer. Try to keep the minimum possible volume 5ul.
- Scale all components proportionally while maintaining valid concentrations (e.g., preserve the master mix factor).
- If you change total reaction volume (e.g., to 40 µL), verify and state that concentrations remain compliant with CONSTANTS.
- Prepare each source on the reagent plate with >=10% overage and SHOW THE MATH.
- Keep the working volume of each well on the plate in mind while designing the reagent plate (e.g., 96-well PCR plate can hold 100ul volume in each well and if you require >= 100ul of any reagent you should have 2 wells on the plate with the reagent).

FINAL PLATE DESIGN
- Exactly 2 Test wells (include Template DNA) and 2 Control wells (exclude Template DNA). Identify explicitly (e.g., A1, A3 = Test; A2, A4 = Control).
- Keep the wells apart on final reaction plate to avoid any contamination.

STRICT STEP FORMAT (no shorthand, no “repeat for wells” language)
X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well] on final plate with 3 mix cycles. [Tip action - eject tip]

OUTPUT STRUCTURE (and nothing else)
1) Materials & Plate Map
2) Liquid Handling Steps (strictly formatted)
3) Thermocycling Steps (3 steps)
4) Fluorescence Read (Hidex; state channel if specified by WebSurfer; otherwise write “Channel not

```
specified by vendor.”)
```

Critique and Validator Prompt

```
You are the combined Critique + Validator agent, part of the PRISM scientific workflow which uses
a multi-agent framework to generate an OT-2-executable Luna Universal qPCR protocol.

Inputs:
- Protocol Planner’s output (Materials & Plate Map, Liquid Handling Steps, Thermocycling, Fluorescence
Read)
- WebSurfer report (CONSTANTS - treat this as the source-of-truth)

--------------------------------
CRITIQUE - CHECKS (enumerate issues with concrete fixes)
--------------------------------

A) CONSISTENCY WITH CONSTANTS
1. Per-reaction component volumes match WebSurfer CONSTANTS OR are proportionally scaled with concentratio
preserved (e.g., maintain 2× master mix factor).
2. Total per-well volume equals the declared total.
3. Any scaling is explicitly justified and keeps primer/template concentrations valid.

B) PLATE MAPS, OVERAGE & CAPACITY
4. Reagent plate lists all sources with >=10% overage and shows arithmetic.
5. Reagent well capacity constraints are respected: if any reagent requires >=100 µL total on a 96-well
PCR plate, it must be split across >=2 source wells with explicit mapping.
6. Final plate has exactly 2 Test wells (include Template DNA) and 2 Control wells (exclude Template
DNA), and the chosen wells are non-adjacent to reduce contamination risk. Identify them explicitly
(e.g., A1, A3 = Test; A2, A4 = Control).

C) TRANSFER MINIMUM, ENUMERATION & TIP HYGIENE
7. No single transfer is < 5 µL. If WebSurfer constants specify sub-5 µL components (e.g., 3 µL primers),
confirm proportional scaling or other compliant solution that keeps concentrations valid.
8. Every liquid-handling instruction is fully enumerated - no “repeat for wells” shorthand.
9. Fresh tips used when touching primers or template; pipette choice is appropriate for the protocol
(e.g., P20 vs P300).

D) FORMAT, LABWARE & DECK
10. **Strict line-format compliance (must match EXACTLY):**
    X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well]
on final plate with 3 mix cycles. [Tip action - eject tip]
    - Volume unit must be µL; reagent names must match Materials/Plate Map.
    - Source wells must exist on the reagent plate; destination wells must exist on the final plate.
11. Deck declarations are explicit and consistent: final plate = slot 1; reagent plate = slot 2;
tip racks = slots 7-8 (or explicitly noted alternatives).

E) THERMOCYCLING & HIDEX
12. Exactly 3 thermocycling steps, matching CONSTANTS in °C and s (no ranges).
13. Hidex read step present; channel specified if in CONSTANTS, else the text “Channel not specified
by vendor.”

If all checks pass: return “No issues found.”
Otherwise: enumerate issues and provide corrected lines and/or corrected sections.

--------------------------------
```

```
VALIDATOR - FINAL OUTPUT
--------------------------------
After applying fixes, return ONLY the final corrected protocol with these sections:
1) Materials & Plate Map
2) Liquid Handling Steps
3) Thermocycling Steps
4) Fluorescence Read

STRICT LIQUID-HANDLING LINE FORMAT (must match EXACTLY)
X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well] on
final plate with 3 mix cycles. [Tip action - eject tip]
- Replace placeholders with concrete values (e.g., “40 µL”, “Luna Universal qPCR Master Mix”, “A1”,
etc.).
- Do not use shorthand or “repeat steps” language.
```

S1.1.3 Cell Painting Prompts

WebSurfer Agent Prompt

```
You are a highly capable agent tasked with surfing the internet to gather comprehensive details about
the PhenoVue™ Cell Painting Kit and its protocol. Your objective is to extract and clearly present:

A detailed list of materials included in the PhenoVue™ Cell Painting Kit.
Step-by-step instructions for preparing staining solutions using the kit reagents.
Protocol steps for cell preparation, staining, fixation, permeabilization, and washing.
Specific guidelines for reagent volumes, incubation times, and handling conditions (e.g., temperature,
light sensitivity).
Recommendations for imaging parameters (e.g., microscope settings for fluorescence channels).
Ensure all information is presented in clear, straightforward English. Avoid technical jargon and
highlight details essential for robotic automation.
```

Protocol Planner Prompt

```
You are an advanced agent tasked with converting a biologically accurate Cell Painting protocol into
a fully OT-2-compatible liquid handling protocol.

Your responsibilities:

1. Convert each step into OT-2-friendly format using:
   - Specific OT-2 slot numbers (1-11)
   - Valid labware types compatible with OT-2 (e.g., 12-column reservoirs, tube racks, 96-well plates)
   - Well positions within labware (e.g., A1, B2, C3)
   - Mix cycles (always 3)
   - Tip actions ("[Tip action: eject]")

2. Labware and labeling:
   - Assign each labware item (e.g., a 96-well plate or reservoir) a **clear, generic label** like
“reaction plate,” “stain reservoir,” or “DNA tube rack.”
   - These labels must be unique and **referenced consistently** throughout the protocol steps.
```

```
   - If multiple items of the same labware type are used (e.g., two 96-well plates), assign them
distinct labels and slot numbers.

3. Reagent and material tracking:
   - Before the protocol steps, generate a **List of Reagents** and a **List of Materials**, with
their assigned OT-2 slots and well positions.
   - In the Materials section, list each labware item with:
     - Slot number
     - Labware type (e.g., 12-channel reservoir, 96-well plate)
     - Assigned label (e.g., “reaction plate”)

4. Reagent volume management:
   - If a reagent is used in multiple steps, reuse the same well until its volume is exhausted.
   - When needed, switch to the next available well or column for that reagent.
   - Track cumulative usage to ensure sufficient volume remains.

Format (MANDATORY) for all pipetting actions:

    X.) Transfer [volume] of [reagent] from [labware label] well [source well] to [labware label]
well [destination well] with 3 mix cycles. [Tip action: eject tip]

Examples:

    1.) Transfer 100 µl of Fixation Buffer from stain reservoir well A3 to reaction plate well B1
with 3 mix cycles. [Tip action: eject]
    2.) Transfer 50 µl of 1X HBSS from wash reservoir well C2 to reaction plate well D2 with 3 mix
cycles. [Tip action: eject]

Notes:
- Do mention labware type and label (e.g., “tube rack,” “reservoir,” “reaction plate”) in the step.
- Do not refer to slots directly in the steps - use the assigned label from the Materials list.
- All steps must be robot-ready, numbered, and fully self-contained.

Your final output must include:
1. A List of Reagents (reagent name + well position + labware label)
2. A List of Materials (slot + labware type + label)
3. A numbered list of OT-2 liquid handling steps (robot-ready and formatted)

This protocol must be biologically accurate, physically feasible, and fully executable on OT-2.
```

Critique Agent Prompt

```
You are tasked with critically analyzing the Cell Painting protocol prepared by the Robotic Agent.
Your objective is to ensure the protocol’s:

- Formatting compliance
- Logical correctness
- Efficiency and optimization for robotic execution
- Proper use of labware labels and reagent/material tracking

Step Formatting:

Each liquid handling instruction must follow this exact format:

    X.) Transfer [volume] of [reagent] from [labware label] well [source well] to [labware label]
well [destination well] with 3 mix cycles. [Tip action: eject tip]
```

```
Example:
    1.) Transfer 50 µl of 1X HBSS from stain reservoir well A1 to reaction plate well B3 with 3 mix
cycles. [Tip action: eject tip]

Invalid Examples:
    - "Add 100 µl of PBS" (missing structure and identifiers)
    - "Transfer cells to new plate" (non-specific, no volumes or positions)
    - "From plate A1 to plate B1" (no reagent name, no labware label)

Your critique tasks:

1. **Reagent and Material Lists**
   - Does the protocol include a complete **List of Reagents** and **List of Materials**?
   - Are reagents properly assigned to wells within labeled labware?
   - Are materials assigned to OT-2 slots with valid labware names and unique labels?

2. **Step Validation**
   - Are all steps properly numbered and formatted?
   - Are reagent names, well positions, and labware labels included and consistent?

3. **Protocol Logic and Flow**
   - Does the sequence follow biological logic for Cell Painting (e.g., staining before fixation)?
   - Are incubation, sealing, and unsealing instructions placed logically?

4. **Optimization Suggestions**
   - Is there unnecessary repetition or redundant transfers?
   - Could the protocol improve pipette efficiency or reduce tip usage?
   - Are reagents reused logically before switching to another well?

Important:
- The format must match robotic standards and be biologically sound.
- Flag issues clearly with bullet points or a numbered list.
- Use robotic-style examples when suggesting improvements.
```

Validator Agent Prompt

```
You are the final validation checkpoint for the Cell Painting protocol generated by the Robotic Agent.

You must confirm the protocol is:
- Robot-compatible
- Logically and structurally sound
- Using valid labware and formatting

---

REQUIRED STRUCTURE:

1. A **List of Reagents** with:
   - Reagent names
   - Assigned labware label
   - Well positions (e.g., A1, B2)

2. A **List of Materials** with:
   - OT-2 slot number
   - Labware type (e.g., 12-column reservoir, 96-well plate)
```

```
   - Assigned label (e.g., stain reservoir, reaction plate)

3. A list of **numbered pipetting steps**, each following this format:

    X.) Transfer [volume] of [reagent] from [labware label] well [source well] to [labware label]
well [destination well] with 3 mix cycles. [Tip action: eject tip]

Examples:
    1.) Transfer 100 µl of Fixation Buffer from stain reservoir well A3 to reaction plate well B1
with 3 mix cycles. [Tip action: eject tip]

Invalid:
    - Steps missing labware labels
    - Vague instructions (e.g., "add buffer to plate")
    - Steps without reagent names or well positions
    - Steps referencing slot numbers instead of labware labels

Your validation tasks:

1. **Format Enforcement**
   - Every step must follow the correct format above.
   - No slot numbers should appear in steps - use labware labels only.

2. **Protocol Logic**
   - Step order should match Cell Painting protocol (reagent prep → staining → fixation → washing
→ imaging)
   - Confirm use of sealing, unsealing, and incubation instructions

3. **Consistency Checks**
   - Every reagent and material referenced in steps must appear in the reagent/material list.
   - Reagents must consistently use the same well until exhausted, then shift to next available well.

Output Instructions:
- If all steps and structure are valid, return the full protocol
- If any requirement is unmet, return:

 - Protocol rejected: Step formatting, logic, or structure is invalid. Please regenerate.

Do not return partial results or explanations. Only validate or reject.
```

S1.2 Single Agent

S1.2.1 Constrained Prompting

```
You are a highly capable reasoning model responsible for generating and verifying a complete, executable
qPCR protocol for the Luna Universal qPCR Master Mix, including both reaction and control wells.

Your objective is to:
1. Gather accurate reagent information and thermocycling details.
2. Generate the step-by-step liquid handling instructions for an Opentrons OT-2 robot.
3. Critically review your own output for accuracy, formatting, and logical feasibility.
4. Return the final corrected and validated protocol ready for execution.
```

```
---

### Reagent Plate Layout (Reaction Plate)
A1-A2: Luna® Universal qPCR Master Mix
C1-C4: Nuclease-free Biowater
D1: Forward Primer
E1: Reverse Primer
H1: Template DNA

---

### Required Specifications
- Reaction wells: A1 and A3 (include Template DNA)
- Control wells: A2 and A4 (exclude Template DNA)
- Volumes (µL):
  • qPCR Master Mix - 20
  • Biowater -  5 for test and 10 for control
  • Forward Primer - 5
  • Reverse Primer - 5
  • Template DNA - 5 (only for A1, A3)
- Total: **18 liquid handling steps** + **3 thermocycling steps**
- Use a 20 µL pipette for all transfers.

---

### Step Format
Each liquid handling step must strictly follow:
‘N.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well]
on final plate with 3 mix cycles. [Tip action - eject tip]‘

Thermocycling steps must follow:
‘N.) Thermocycling: [Temperature] for [Duration].‘

---

### Self-Check Before Final Output
After generating the protocol, verify:
- All reagent volumes and well mappings are correct.
- Reaction wells include Template DNA; control wells exclude it.
- Exactly 18 liquid-handling and 3 thermocycling steps exist.
- Every step follows the exact required format.
- Steps are ordered logically and feasible for OT-2 execution.

---

### Output Format
Produce only the final validated protocol, structured as:

1. Materials
2. Liquid Handling Steps
3. Thermocycling Steps
4. Absorbance Reading (using Hidex plate reader)

Avoid melt-curve analysis. Keep all numbers explicit (no ranges). Do not include commentary or reasoning
in the final answer.
```

S1.2.2 Open-Ended Prompting

```
You are a highly capable reasoning model responsible for producing a complete, OT-2-executable qPCR
protocol using the Luna Universal qPCR Master Mix.
Your job is to (A) retrieve authoritative protocol constants, and (B) convert them into an OT-2-compatible
protocol that fits the lab setup below. Produce only the final validated protocol. The generated
protocol must be **physically feasible on an OT-2** and **biologically sane** (maintain correct concentrat
avoid inhibitory conditions, and minimize cross-contamination).

GOALS
1) Retrieve the official Luna Universal qPCR Master Mix protocol and extract required constants:
component list, per-reaction volumes, thermocycling temperatures/times, and dye compatibility.
2) Generate explicit, enumerated OT-2 liquid-handling and thermocycling steps that satisfy the LAB
SETUP & CONSTRAINTS.
3) Include two TEST and two CONTROL reactions placed far apart to minimize cross-contamination.
4) End with a fluorescence read on the Hidex plate reader.

LAB SETUP & CONSTRAINTS
- OT-2 deck: **final PCR plate** in **Slot 1**, **reagent plate** in **Slot 2**.
- Use **96-well PCR plates** for both.
- **Working volume per well:** 100 µL. If total reagent use exceeds this, split it across multiple
source wells and note it in the Plate Map.
- **Minimum single aspiration/dispense:** 5 µL - no transfers below this volume.
- **Forward and reverse primers must be handled separately.** Each primer requires its own transfer
step (>= 5 µL) per reaction well;
- **Reaction volume for TEST wells:** 40 µL (derived from Luna constants).
- **CONTROL wells:** identical setup but omit template DNA (replace with proper diluent).
- **Thermocycling:** exactly 3 steps total using Luna’s official stage temperatures/durations.
- **Fluorescence read:** Hidex plate reader, SYBR/FAM channel.
- Do **not** use “repeat steps”; enumerate every transfer explicitly.
- Assign source wells yourself and report them in the Plate Map.
- Include any necessary overage/dead-volume adjustments in the reagent table.

FORMAT RULES
Produce only these six sections in order - no commentary:
  1. Materials
  2. Plate Maps
  3. Reagent Preparation (include overage math if used)
  4. Liquid Handling Steps
  5. Thermocycling Steps
  6. Fluorescence Read

Each liquid-handling instruction must follow this exact format:
  N.) Transfer [volume] of [reagent] from [source well] on reagent plate to [destination well] on
final plate with 3 mix cycles. [Tip action - eject tip]

Thermocycling format:
  N.) Thermocycling: [TEMPERATURE °C] for [DURATION s].

Fluorescence read (one line):
  N.) Fluorescence read: Hidex plate reader, SYBR/FAM channel.
```

PLANNING REQUIREMENTS (INTERNAL - DO NOT PRINT)
- Compute per-component volumes from official Luna constants so TEST = 40 µL.
- Choose two TEST and two CONTROL wells with maximal spatial separation; list them in the Plate Map.
- If any reagent’s total use exceeds 100 µL, split across multiple source wells and record mapping.
- Verify 5 µL minimum per aspiration, proper reagent-to-well mappings, correct inclusion/exclusion of template, and formatting compliance.
- All steps must be executable with standard OT-2 pipettes (fresh tip per transfer, correct mixing).
- Output must contain exactly the six sections above and nothing else.

S1.2.3 Cell Painting Prompts

You are a single high-level reasoning model responsible for generating a complete, executable Cell Painting protocol using the PhenoVue™ Cell Painting Kit, designed for Opentrons OT-2 automation.

You will think and act holistically - not as multiple agents - to plan, reason through, and output the final validated protocol ready for robotic execution.

OBJECTIVE:

Produce a self-contained OT-2-ready protocol that is biologically correct, physically feasible, and formatted for robotic execution.

TASKS

1. UNDERSTAND THE TASK
   - Recall or infer all necessary details about the PhenoVue™ Cell Painting Kit: reagents, staining and fixation solutions, incubation steps, and imaging readout.
   - Use reasoning to fill in realistic but clearly stated assumptions for reagent volumes, incubation times, or handling steps if exact data are not available.

2. DESIGN THE ROBOTIC WORKFLOW
   - Assign OT-2 labware to slots (1-11) and label them clearly (e.g., ‘‘reaction plate,’’ ‘‘stain reservoir,’ ‘‘wash reservoir’’).
   - Plan the sequence of operations following the logical Cell Painting workflow:
     cell seeding → staining → fixation → permeabilization → washing → imaging.
   - Ensure all transfers, incubations, and sealing/unsealing actions follow a practical robotic order.

3. GENERATE THE STRUCTURED OUTPUT
   The final protocol must include:
   - List of Reagents: reagent name + labware label + well position(s)
   - List of Materials: OT-2 slot + labware type + assigned label
   - Liquid Handling Steps: numbered and formatted exactly as:

   X.) Transfer [volume] of [reagent] from [Well on plate] on reaction plate to [destination well] on final plate with 3 mix cycles. [Tip action - eject tip]

4. REASON THROUGH VALIDATION
   - Check your own output for:
     - Correct structure (Reagents, Materials, Steps)

```
     • Consistency between lists and steps
     • Logical biological order and feasibility
     • Proper format (no slot numbers in steps, only labels)
   - If an inconsistency is found, fix it before final output.

5. DELIVER ONLY THE FINAL, VALIDATED PROTOCOL
   - Do not describe your reasoning process.
   - Output only the final, fully verified protocol, ready for OT-2 execution.

EXPECTED OUTPUT FORMAT
1. List of Reagents
2. List of Materials
3. Liquid Handling Steps (numbered)
4. Incubation / Sealing / Imaging Notes (if relevant)
```

S2 Protocol Generation (Stage 2)

S2.1 Initial Protocol Generation Prompt

The following is the full prompt used in the Protocol Generation / Validation section of the main article for PRISM to convert the English-language plans into robot-compatible steps for the PCR protocol. It covers the available robots in the laboratory, the desired output formats, and ends with the specific workflow to be converted.

```
# YAML Workflow File Documentation

## Modules and Actions

### Module: ‘biopf400‘
**Purpose**: Robotic transfer system (main plate handling robot)

#### Access:
This module does not have its own locations, and instead facilitates transfers between all other
locations.

#### Restrictions:
- ‘transfer‘ requires that a plate is present at the source location
- ‘transfer‘ requires that the target location is available/empty
- ‘transfer‘ can only occur between locations with the same plate rotation (both narrow or both wide)
- If a transfer needs to change plate orientation (narrow to wide or wide to narrow), it must first
go through the exchange station as an intermediate step

#### Actions:
- **‘transfer‘**: Moves plates between different locations/stations with configurable positioning
  - **Arguments**:
    - ‘source‘: Source location identifier
    - ‘source_approach‘: Approach path identifier for source pickup
    - ‘source_plate_rotation‘: Plate orientation at source (‘wide‘ or ‘narrow‘)
    - ‘target‘: Target location identifier
    - ‘target_approach‘: Approach path identifier for target placement
    - ‘target_plate_rotation‘: Plate orientation at target (‘wide‘ or ‘narrow‘)

### Module: ‘ot2bioalpha‘
**Purpose**: OT-2 liquid handling robot

#### Access:
**Locations:**
- ‘ot2bioalpha_deck1_wide‘ - OT-2 robot deck position 1 (wide)

**Approach Paths:**
- ‘safe_path_ot2bioalpha‘ - Safe approach path for OT-2 robot

#### Restrictions:
- ‘run_protocol‘ can only be executed when plates and consumables are properly loaded on the deck

#### Actions:
- **‘run_protocol‘**: Executes a liquid handling protocol using a specified protocol file
  - **Arguments**:
    - ‘files‘: Dictionary containing protocol files
    - ‘protocol‘: Path to the OT-2 protocol file (e.g., ‘payload.ot2_protocol‘)

### Module: ‘bio_sealer‘
```

```
**Purpose**: Plate sealing station

#### Access:
**Locations:**
- ‘sealer_nest‘ - Plate sealing station nest (narrow)

**Approach Paths:**
- ‘safe_path_sealer‘ - Safe approach path for sealing station

#### Restrictions:
- ‘seal‘ can only be executed when a plate is present in the sealer nest

#### Actions:
- **‘seal‘**: Seals plates (typically with adhesive film or heat seal)
  - **Arguments**: None (empty args)

### Module: ‘bio_biometra3_96‘
**Purpose**: Biometra thermocycler for PCR

#### Access:
**Locations:**
- ‘bio_biometra3_nest‘ - Biometra thermocycler nest (wide)

**Approach Paths:**
- ‘safe_path_biometra3‘ - Safe approach path for Biometra thermocycler

#### Restrictions:
- Must be ‘open‘ before a plate can be placed in or removed from the nest
- Must be ‘close‘d before ‘run_program‘ can be executed
- ‘run_program‘ can only be executed when closed and with a plate present in the nest
- Must be ‘open‘ed again after ‘run_program‘ completes to remove the plate

#### Actions:
- **‘open‘**: Opens the thermocycler lid
  - **Arguments**: None (empty args)
- **‘close‘**: Closes the thermocycler lid
  - **Arguments**: None (empty args)
- **‘run_program‘**: Executes a pre-programmed thermocycling protocol
  - **Arguments**:
    - ‘program_number‘: Integer specifying which stored program to run (e.g., ‘5‘)

### Module: ‘bio_peeler‘
**Purpose**: Plate peeling/unsealing station

#### Access:
**Locations:**
- ‘peeler_nest‘ - Plate peeling station nest (narrow)

**Approach Paths:**
- ‘safe_path_peeler‘ - Safe approach path for peeling station

#### Restrictions:
- ‘peel‘ can only be executed when a sealed plate is present in the peeler nest

#### Actions:
- **‘peel‘**: Removes seals from plates
```

```
  - **Arguments**: None (empty args)

### Module: ‘hidex_geraldine‘
**Purpose**: Hidex plate reader for assays/measurements

#### Access:
**Locations:**
- ‘hidex_geraldine_high_nest‘ - Hidex plate reader high nest position (narrow)

**Approach Paths:**
- ‘safe_path_hidex‘ - Safe approach path for Hidex plate reader

#### Restrictions:
- Must be ‘open‘ before a plate can be placed in or removed from the nest
- Must be ‘close‘d before ‘run_assay‘ can be executed
- ‘run_assay‘ can only be executed when closed and with a plate present in the nest
- Must be ‘open‘ed again after ‘run_assay‘ completes to remove the plate

#### Actions:
- **‘open‘**: Opens the plate reader lid
  - **Arguments**: None (empty args)
- **‘close‘**: Closes the plate reader lid
  - **Arguments**: None (empty args)
- **‘run_assay‘**: Runs a specified assay protocol
  - **Arguments**:
    - ‘assay_name‘: String name of the assay protocol (e.g., ‘"PCR_Final_Results"‘)

### Exchange Station
**Purpose**: The exchange station serves as an intermediate transfer point between modules, allowing
for plate orientation changes and temporary storage during multi-step workflows.

#### Access:
**Locations:**
- ‘exchange_deck_high_wide‘ - Exchange station high position (wide)
- ‘exchange_deck_high_narrow‘ - Exchange station high position (narrow)

**Approach Paths:**
- ‘safe_path_exchange‘ - Safe approach path for exchange station

#### Restrictions:
- The exchange deck must either be a source or target location in a transfer, NEVER both
- The used exchange deck orientation must match the other location, being either both wide or both
narrow
- Transfers via the exchange deck should use 2 transfers, the first having the exchange as the target,
and the second having the exchange as the source

# YAML Workflow File Format Specification

This section describes the required format and structure for YAML workflow files used in automated
laboratory protocols.

## Top-Level Structure

Every YAML workflow file must contain the following top-level fields followed by a ‘flowdef‘ section:

‘‘‘yaml
```

```
name: [Protocol Name]
author: [Author/Organization]
info: [Description of the protocol]
version: ’[Version Number]’

flowdef:
  # List of actions goes here
‘‘‘

## Metadata Fields

### ‘name‘ (Required)
- **Type**: String
- **Purpose**: Human-readable name of the protocol
- **Example**: ‘"Test Protocol"‘, ‘"PCR Amplification Workflow"‘

### ‘author‘ (Required)
- **Type**: String
- **Purpose**: Name of the person or organization that created the protocol
- **Example**: ‘"Autoprotocol"‘, ‘"Lab Automation Team"‘

### ‘info‘ (Required)
- **Type**: String
- **Purpose**: Brief description of what the protocol accomplishes
- **Example**: ‘"A PCR protocol written by Autoprotocol"‘, ‘"Automated qPCR workflow for gene expression
analysis"‘

### ‘version‘ (Required)
- **Type**: String (quoted)
- **Purpose**: Version identifier for protocol tracking and updates
- **Example**: ‘’0.1’‘, ‘’1.2.3’‘, ‘’2024.01’‘

## Flow Definition (‘flowdef‘)

The ‘flowdef‘ section contains an ordered list of actions that define the workflow execution sequence.

### Action Structure

Each action in the ‘flowdef‘ list must have the following structure:

‘‘‘yaml
  - action: [action_name]
    name: [human_readable_description]
    module: [module_identifier]
    args: [arguments_object]        # Optional
    files: [files_object]           # Optional
‘‘‘

### Action Fields

#### ‘action‘ (Required)
- **Type**: String
- **Purpose**: Specifies which action to perform
- **Valid Values**: See module documentation for available actions
- **Examples**: ‘run_protocol‘, ‘transfer‘, ‘seal‘, ‘open‘, ‘close‘, ‘run_program‘, ‘peel‘, ‘run_assay‘
```

```
#### ‘name‘ (Required)
- **Type**: String
- **Purpose**: Human-readable description of what this specific action does
- **Example**: ‘"Run liquid protocol"‘, ‘"Transfer the destination plate from OT-2 to exchange"‘

#### ‘module‘ (Required)
- **Type**: String
- **Purpose**: Identifies which equipment module will execute the action
- **Valid Values**: ‘ot2bioalpha‘, ‘biopf400‘, ‘bio_sealer‘, ‘bio_biometra3_96‘, ‘bio_peeler‘, ‘hidex_geral

#### ‘args‘ (Optional)
- **Type**: Object/Dictionary
- **Purpose**: Contains arguments specific to the action being performed
- **Usage**: Required for actions that need parameters (e.g., transfer locations, program numbers)
- **Example**:
‘‘‘yaml
    args:
      source: ot2bioalpha_deck1_wide
      target: exchange_deck_high_wide
      source_approach: safe_path_ot2bioalpha
      target_approach: safe_path_exchange
      source_plate_rotation: wide
      target_plate_rotation: wide
‘‘‘

#### ‘files‘ (Optional)
- **Type**: Object/Dictionary
- **Purpose**: Specifies file references needed for the action
- **Usage**: Used primarily with ‘run_protocol‘ actions
- **Example**:
‘‘‘yaml
    files:
      protocol: payload.ot2_protocol
‘‘‘

## Complete Example

‘‘‘yaml
name: Sample Protocol
author: PRISM
info: Automated workflow with sealing
version: ’1.0’

flowdef:
  - action: run_protocol
    name: Execute liquid handling protocol
    module: ot2bioalpha
    files:
      protocol: payload.ot2_protocol

  - action: transfer
    name: Move plate from OT-2 to sealer
    module: biopf400
    args:
      source: ot2bioalpha_deck1_wide
      source_approach: safe_path_ot2bioalpha
```

```
      source_plate_rotation: wide
      target: sealer_nest
      target_approach: safe_path_sealer
      target_plate_rotation: narrow

  - action: seal
    name: Seal the reaction plate
    module: bio_sealer
    args: {}
‘‘‘

## Important Notes

- **YAML Syntax**: Ensure proper indentation (2 spaces recommended) and valid YAML formatting
- **Action Order**: Actions execute sequentially in the order listed
- **Empty Args**: Use ‘args: {}‘ for actions that require the args field but take no parameters
- **Comments**: Use ‘#‘ for comments and section dividers to improve readability
- **String Quoting**: Quote version numbers and strings containing special characters

# OT-2 Liquid Handling File Format Specification

This document describes the required format and structure for Python protocol files used with the
Opentrons OT-2 liquid handling robot.

## Top-Level Structure

Every OT-2 protocol file must contain the following components:

‘‘‘python
requirements = {"robotType": "OT-2"}
from opentrons import protocol_api

metadata = {
    "protocolName": "PCR",                     # Human-readable protocol name
    "author": "PRISM",                         # Protocol author/creator
    "description": "PCR",                      # Brief protocol description
    "apiLevel": "2.12",                        # Opentrons API version
    "info": "A PCR protocol written by PRISM", # Additional information
    "name": "PCR",                             # Short protocol name
    "version": "1.0"                           # Version identifier
}

def run(protocol: protocol_api.ProtocolContext):
    # Protocol implementation
‘‘‘

## Protocol Function

### Module Loading
‘‘‘python
module = protocol.load_module("Temperature Module", "3")
deck["3"] = module.load_labware("nest_96_wellplate_100ul_pcr_full_skirt")
deck["3"].set_offset(x=1.0, y=1.4, z=5.7)
‘‘‘

### Standard Labware Loading
```

```
‘‘‘python
deck["1"] = protocol.load_labware("nest_96_wellplate_100ul_pcr_full_skirt", "1")
deck["1"].set_offset(x=0.9, y=-0.5, z=0.6)
‘‘‘

### Tip Rack Loading
‘‘‘python
deck["7"] = protocol.load_labware("opentrons_96_tiprack_20ul", "7")
deck["7"].set_offset(x=0.2, y=1.6, z=-0.8)
‘‘‘

### Pipette Loading
‘‘‘python
pipettes["left"] = protocol.load_instrument(
    "p20_single_gen2",
    "left",
    tip_racks=[deck["7"]]
)
‘‘‘

### Standard Transfer Pattern

The basic pattern for a single transfer operation with mixing:

‘‘‘python
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(20.0, deck["3"]["A1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(20.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()
‘‘‘

### Operation Breakdown

1. **‘pick_up_tip()‘**: Retrieves a new tip from the tip rack
2. **‘well_bottom_clearance.aspirate‘**: Sets height above well bottom for aspiration (in mm)
3. **‘aspirate(volume, location)‘**: Draws liquid from specified well
4. **‘well_bottom_clearance.dispense‘**: Sets height above well bottom for dispensing (in mm)
5. **‘dispense(volume, location)‘**: Dispenses liquid into specified well
6. **‘mix(repetitions, volume, location)‘**: Mixes by aspirating and dispensing repeatedly
7. **‘blow_out()‘**: Expels any remaining liquid from the tip
8. **‘drop_tip()‘**: Discards the used tip

### Mix-Only Operation

For mixing without transfer:

‘‘‘python
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].mix(10, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
```

```
pipettes["left"].drop_tip()
‘‘‘

## Minimal Working Example

‘‘‘python
requirements = {"robotType": "OT-2"}
from opentrons import protocol_api

metadata = {
    "protocolName": "Simple Transfer",
    "author": "PRISM",
    "description": "Basic liquid transfer example",
    "apiLevel": "2.12",
    "info": "Demonstrates minimal OT-2 protocol structure",
    "name": "Simple Transfer",
    "version": "1.0"
}

def run(protocol: protocol_api.ProtocolContext):
    deck = {}
    pipettes = {}

    #################
    # load labware #
    #################
    deck["1"] = protocol.load_labware("nest_96_wellplate_100ul_pcr_full_skirt", "1")
    deck["2"] = protocol.load_labware("nest_96_wellplate_100ul_pcr_full_skirt", "2")
    deck["7"] = protocol.load_labware("opentrons_96_tiprack_20ul", "7")
    pipettes["left"] = protocol.load_instrument("p20_single_gen2", "left", tip_racks=[deck["7"]])

    ####################
    # execute commands #
    ####################
    # Transfer 10 µL from source to destination with mixing
    pipettes["left"].pick_up_tip()
    pipettes["left"].well_bottom_clearance.aspirate = 1
    pipettes["left"].aspirate(10.0, deck["2"]["A1"])
    pipettes["left"].well_bottom_clearance.dispense = 1
    pipettes["left"].dispense(10.0, deck["1"]["A1"])
    pipettes["left"].mix(3, 10, deck["1"]["A1"])
    pipettes["left"].blow_out()
    pipettes["left"].drop_tip()
‘‘‘

## Important Notes

- **Tip Usage**: Each transfer operation uses a fresh tip (pick_up_tip at start, drop_tip at end)
- **Well References**: Use string notation for well positions (e.g., ‘"A1"‘, ‘"B2"‘)
- **Comments**: Use ‘#‘ for comments and section dividers to improve readability
- **Sequential Execution**: Commands execute in the order written

# PCR Workflow Sequence Overview

These are the high-level workflow steps for an automated PCR processing experiment.
```

The ‘ot2bioalpha‘ OT-2 is prepared with the 96-well plates ‘react_plate‘ on a Temperature Module at deck position 3 and ‘dest_plate‘ at deck position 1

1.) Transfer 20 uL of qPCR Master Mix from react_plate to dest_plate: A1→B2, A2→B11, A1→G2, A2→G11 with 3 mix cycles. [Tip action - eject]
2.) Transfer 5 uL of Nuclease-free Biowater from react_plate to dest_plate: C1→B2, C2→B11, C3→G2, C4→G11 with 3 mix cycles. [Tip action - eject]
3.) Transfer 5 uL of Forward Primer from react_plate D1 to dest_plate wells B2, B11, G2, and G11 with 3 mix cycles. [Tip action - eject]
4.) Transfer 5 uL of Reverse Primer from react_plate E1 to dest_plate wells B2, B11, G2, and G11 with 3 mix cycles. [Tip action - eject]
5.) Transfer 5 uL of Template DNA from react_plate H1 to dest_plate wells B2 and G2 with 3 mix cycles. [Tip action - eject]
6.) Transfer 5 uL of Nuclease-free Biowater from react_plate to dest_plate: C2→B11, C4→G11 with 3 mix cycles. [Tip action - eject]
7.) Mix 10X with pipette set to 20 uL in wells B2, G2, B11, and G11
8.) Seal the plate to prepare it for thermocycling
9.) Perform a thermocycling operation (program #5)
10.) Peel the plate to prepare it for reading
11.) Perform an absorbance reading using the Hidex plate reader (protocol "PCR_Final_Results")
12.) Open the Hidex plate reader lid after analysis completion

---

You are PRISM, an automated protocol designer for scientific workflows.
Using the provided reference material, create all OT-2 python files and workflow yaml files for the PCR workflow sequence.
Group all adjacent liquid transfer operations into a single execution of a liquid handling protocol on the OT-2 robot. Come up with a unique and descriptive name for each OT-2 protocol Python file.
All steps outside of liquid handling, including any in comments or described as manual, should be handled in the yaml file.

S2.2 Initial Protocol Generation Output

The following is the raw output from Claude Sonnet 4.5 on the prompt in the previous section. Note that the raw output describes two files (a Python file and a YAML file), which were manually extracted and used in the downstream tasks.

```
Looking at the PCR workflow sequence, I’ll create the necessary files. Steps 1-7 are liquid handling
operations that should be grouped into a single OT-2 protocol, while steps 8-12 involve other modules
in the YAML workflow.

# File 1: pcr_reaction_setup.ot2_protocol

‘‘‘python
requirements = {"robotType": "OT-2"}
from opentrons import protocol_api

metadata = {
    "protocolName": "PCR Reaction Setup",
    "author": "PRISM",
    "description": "Complete PCR reaction assembly with master mix, primers, template DNA, and controls",
    "apiLevel": "2.12",
    "info": "Automated PCR reaction setup protocol",
    "name": "PCR Reaction Setup",
    "version": "1.0"
}

def run(protocol: protocol_api.ProtocolContext):
    deck = {}
    pipettes = {}

    ################
    # load labware #
    ################
    # Load Temperature Module with react_plate at position 3
    module = protocol.load_module("Temperature Module", "3")
    deck["3"] = module.load_labware("nest_96_wellplate_100ul_pcr_full_skirt")
    deck["3"].set_offset(x=1.0, y=1.4, z=5.7)

    # Load dest_plate at position 1
    deck["1"] = protocol.load_labware("nest_96_wellplate_100ul_pcr_full_skirt", "1")
    deck["1"].set_offset(x=0.9, y=-0.5, z=0.6)

    # Load tip rack
    deck["7"] = protocol.load_labware("opentrons_96_tiprack_20ul", "7")
    deck["7"].set_offset(x=0.2, y=1.6, z=-0.8)

    pipettes["left"] = protocol.load_instrument(
        "p20_single_gen2",
        "left",
        tip_racks=[deck["7"]]
    )

    ####################
    # execute commands #
    ####################
```

```
# Step 1: Transfer 20 uL of qPCR Master Mix
# A1→B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(20.0, deck["3"]["A1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(20.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# A2→B11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(20.0, deck["3"]["A2"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(20.0, deck["1"]["B11"])
pipettes["left"].mix(3, 20, deck["1"]["B11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# A1→G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(20.0, deck["3"]["A1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(20.0, deck["1"]["G2"])
pipettes["left"].mix(3, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# A2→G11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(20.0, deck["3"]["A2"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(20.0, deck["1"]["G11"])
pipettes["left"].mix(3, 20, deck["1"]["G11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 2: Transfer 5 uL of Nuclease-free Biowater
# C1→B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# C2→B11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C2"])
```

```
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B11"])
pipettes["left"].mix(3, 20, deck["1"]["B11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# C3→G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C3"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G2"])
pipettes["left"].mix(3, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# C4→G11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C4"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G11"])
pipettes["left"].mix(3, 20, deck["1"]["G11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 3: Transfer 5 uL of Forward Primer from D1 to all destinations
# D1→B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["D1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# D1→B11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["D1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B11"])
pipettes["left"].mix(3, 20, deck["1"]["B11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# D1→G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["D1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G2"])
pipettes["left"].mix(3, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()
```

```
# D1→G11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["D1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G11"])
pipettes["left"].mix(3, 20, deck["1"]["G11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 4: Transfer 5 uL of Reverse Primer from E1 to all destinations
# E1→B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["E1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# E1→B11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["E1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B11"])
pipettes["left"].mix(3, 20, deck["1"]["B11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# E1→G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["E1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G2"])
pipettes["left"].mix(3, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# E1→G11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["E1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G11"])
pipettes["left"].mix(3, 20, deck["1"]["G11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 5: Transfer 5 uL of Template DNA from H1 to sample wells (not controls)
# H1→B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
```

```python
pipettes["left"].aspirate(5.0, deck["3"]["H1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B2"])
pipettes["left"].mix(3, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# H1→G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["H1"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G2"])
pipettes["left"].mix(3, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 6: Transfer 5 uL of Nuclease-free Biowater to control wells (instead of template)
# C2→B11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C2"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["B11"])
pipettes["left"].mix(3, 20, deck["1"]["B11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# C4→G11
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].aspirate(5.0, deck["3"]["C4"])
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].dispense(5.0, deck["1"]["G11"])
pipettes["left"].mix(3, 20, deck["1"]["G11"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Step 7: Final mixing - 10 cycles with 20 uL in all destination wells
# Mix B2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].mix(10, 20, deck["1"]["B2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Mix G2
pipettes["left"].pick_up_tip()
pipettes["left"].well_bottom_clearance.aspirate = 1
pipettes["left"].well_bottom_clearance.dispense = 1
pipettes["left"].mix(10, 20, deck["1"]["G2"])
pipettes["left"].blow_out()
pipettes["left"].drop_tip()

# Mix B11
```

```
    pipettes["left"].pick_up_tip()
    pipettes["left"].well_bottom_clearance.aspirate = 1
    pipettes["left"].well_bottom_clearance.dispense = 1
    pipettes["left"].mix(10, 20, deck["1"]["B11"])
    pipettes["left"].blow_out()
    pipettes["left"].drop_tip()

    # Mix G11
    pipettes["left"].pick_up_tip()
    pipettes["left"].well_bottom_clearance.aspirate = 1
    pipettes["left"].well_bottom_clearance.dispense = 1
    pipettes["left"].mix(10, 20, deck["1"]["G11"])
    pipettes["left"].blow_out()
    pipettes["left"].drop_tip()
‘‘‘

# File 2: pcr_workflow.yaml

‘‘‘yaml
name: Automated PCR Workflow
author: PRISM
info: Complete automated PCR workflow including reaction setup, thermocycling, and plate reading
version: ’1.0’

flowdef:
  #########################
  # Liquid Handling Setup #
  #########################
  - action: run_protocol
    name: Execute PCR reaction setup protocol
    module: ot2bioalpha
    files:
      protocol: payload.pcr_reaction_setup.ot2_protocol

  ################################
  # Transfer to Sealer (via Exchange)
  ################################
  - action: transfer
    name: Transfer plate from OT-2 to exchange station
    module: biopf400
    args:
      source: ot2bioalpha_deck1_wide
      source_approach: safe_path_ot2bioalpha
      source_plate_rotation: wide
      target: exchange_deck_high_wide
      target_approach: safe_path_exchange
      target_plate_rotation: wide

  - action: transfer
    name: Transfer plate from exchange to sealer with orientation change
    module: biopf400
    args:
      source: exchange_deck_high_narrow
      source_approach: safe_path_exchange
      source_plate_rotation: narrow
      target: sealer_nest
```

```
    target_approach: safe_path_sealer
    target_plate_rotation: narrow

##########
# Sealing
##########
- action: seal
  name: Seal the PCR reaction plate
  module: bio_sealer
  args: {}

###################################
# Transfer to Thermocycler (via Exchange)
###################################
- action: transfer
  name: Transfer sealed plate from sealer to exchange station
  module: biopf400
  args:
    source: sealer_nest
    source_approach: safe_path_sealer
    source_plate_rotation: narrow
    target: exchange_deck_high_narrow
    target_approach: safe_path_exchange
    target_plate_rotation: narrow

- action: transfer
  name: Transfer plate from exchange to thermocycler with orientation change
  module: biopf400
  args:
    source: exchange_deck_high_wide
    source_approach: safe_path_exchange
    source_plate_rotation: wide
    target: bio_biometra3_nest
    target_approach: safe_path_biometra3
    target_plate_rotation: wide

###############
# Thermocycling
###############
- action: open
  name: Open thermocycler lid
  module: bio_biometra3_96
  args: {}

- action: close
  name: Close thermocycler lid
  module: bio_biometra3_96
  args: {}

- action: run_program
  name: Execute PCR thermocycling program
  module: bio_biometra3_96
  args:
    program_number: 5

- action: open
```

```
  name: Open thermocycler lid after completion
  module: bio_biometra3_96
  args: {}

###############################
# Transfer to Peeler (via Exchange)
###############################
- action: transfer
  name: Transfer plate from thermocycler to exchange station
  module: biopf400
  args:
    source: bio_biometra3_nest
    source_approach: safe_path_biometra3
    source_plate_rotation: wide
    target: exchange_deck_high_wide
    target_approach: safe_path_exchange
    target_plate_rotation: wide

- action: transfer
  name: Transfer plate from exchange to peeler with orientation change
  module: biopf400
  args:
    source: exchange_deck_high_narrow
    source_approach: safe_path_exchange
    source_plate_rotation: narrow
    target: peeler_nest
    target_approach: safe_path_peeler
    target_plate_rotation: narrow

##########
# Peeling
##########
- action: peel
  name: Remove seal from PCR plate
  module: bio_peeler
  args: {}

####################################
# Transfer to Plate Reader (via Exchange)
####################################
- action: transfer
  name: Transfer peeled plate from peeler to exchange station
  module: biopf400
  args:
    source: peeler_nest
    source_approach: safe_path_peeler
    source_plate_rotation: narrow
    target: exchange_deck_high_narrow
    target_approach: safe_path_exchange
    target_plate_rotation: narrow

- action: transfer
  name: Transfer plate from exchange to Hidex plate reader
  module: biopf400
  args:
    source: exchange_deck_high_narrow
```

```
      source_approach: safe_path_exchange
      source_plate_rotation: narrow
      target: hidex_geraldine_high_nest
      target_approach: safe_path_hidex
      target_plate_rotation: narrow

  ###############
  # Plate Reading
  ###############
  - action: open
    name: Open Hidex plate reader lid
    module: hidex_geraldine
    args: {}

  - action: close
    name: Close Hidex plate reader lid
    module: hidex_geraldine
    args: {}

  - action: run_assay
    name: Run PCR final results assay
    module: hidex_geraldine
    args:
      assay_name: "PCR_Final_Results"

  - action: open
    name: Open Hidex plate reader lid after analysis
    module: hidex_geraldine
    args: {}
‘‘‘

These files implement the complete PCR workflow:

**pcr_reaction_setup.ot2_protocol**: Groups all liquid handling steps (1-7) into a single OT-2 protocol
that:
- Adds qPCR Master Mix to all destination wells
- Adds nuclease-free water
- Adds forward and reverse primers to all wells
- Adds template DNA to sample wells (B2, G2)
- Adds water to control wells (B11, G11) instead of template
- Performs final mixing of all wells

**pcr_workflow.yaml**: Orchestrates the entire workflow including:
- Running the liquid handling protocol
- Transferring plates between modules using the exchange station for orientation changes
- Sealing, thermocycling, peeling, and plate reading operations
- Following all module restrictions (opening/closing lids, proper plate orientations)
```

S2.3 Iteration with Simulation Feedback Prompt

The following is the full response given to the Claude Sonnet 4.5 model based on the simulation results of the previous section's output, which prompts Claude to identify and fix the error. This process is repeated with each new error until the simulation detects no errors.

```
Error while executing step 6 (‘action=transfer, module=biopf400, args={’source’: ’exchange_deck_high_wide’
’source_approach’: ’safe_path_exchange’, ’source_plate_rotation’: ’wide’, ’target’: ’bio_biometra3_nest’,
’target_approach’: ’safe_path_biometra3’, ’target_plate_rotation’: ’wide’}‘): Collision between microplate
and bio_biometra3_96
Rewrite the full YAML file with the correction applied.
```