Contents

1 Observation vs Intervention: $p(x \mid c)$ vs $p(x \mid do(c))$

The difference between *observing* and *intervening* is the foundational distinction in causal inference. It separates "what correlates with what" from "what causes what"—and it's the fault line between classical machine learning and causal ML.

This document explains why this distinction matters enormously for computational biology, where the goal is often to find *actionable* targets, not just *predictive* markers.

1.1 Table of Contents

1. The Observational Quantity
2. The Interventional Quantity
3. Why These Two Are Not the Same
4. Conditioning vs Intervention
5. Why This Matters for Computational Biology
6. What Comes Next

1.2 The Observational Quantity

When you write

> $p(x \mid c)$ — the probability that $X = x$ given condition c

you are living entirely inside the world of **observations**.

You looked at data where the condition *happened to be* c, and you summarized what X looked like *in those cases.*

In words, this reads as:

> "Among samples where the condition was c, how often do I see x?"

That's it. No claims about causality are being made — only association.

1.2.1 A Biological Mental Picture

Imagine:

- c: "cell expresses high MYC"

- x: “cell shows increased ribosomal gene expression”

If you compute $p(x \mid c)$, you’re saying:

> “Among cells with high MYC, what fraction have high ribosomal genes?”

This is descriptive. It’s correlational. It’s passive.

Critically:

- You **did not intervene**
- You **did not force MYC to be high**
- You **did not block confounders**

The world did whatever it wanted, and you took notes.

1.2.2 Why ML Loves This World

Most of machine learning lives here because it’s convenient:

- Supervised learning estimates $p(y \mid x)$
- VAEs estimate $p(x \mid z)$
- Classifiers, regressors, density estimators — all **observational**

They’re excellent at pattern recognition. They are silent on **why**.

1.3 The Interventional Quantity

Now we cross the Rubicon.

When you write

> $p(x \mid do(c))$

you are no longer describing the world — you are **imagining breaking it and rewiring it**.

The $do(\cdot)$ operator means:

> “I *set* c to a value, ignoring whatever would normally cause c.”

This is not conditioning. This is **surgery**.

In words:

> “If I were to force c to this value, what would X become?”

That single word *force* is the whole game.

1.3.1 Biological Mental Picture

Same variables:

- c: MYC expression
- x: ribosomal genes

Now $p(x \mid do(c = \text{high}))$ means:

> “If I experimentally overexpress MYC — CRISPRa, viral vector, whatever — what happens to ribosomal gene expression?”

This is the language of:

- Perturb-seq
- CRISPR screens

- Drug treatments
- Knockdowns and overexpression

This is causal biology.

1.4 Why These Two Are *Not* the Same

Here's the key truth that causes most confusion:

> **In general:**
>
> $$p(x \mid c) \neq p(x \mid do(c))$$

They are equal only under strong assumptions — essentially, *no confounding*.

1.4.1 The Confounder Problem

Imagine a hidden variable:

- z: cell cycle state

Cell cycle:

- Pushes MYC high
- Independently pushes ribosomal genes high

Now observationally:

- High MYC and high ribosomal expression co-occur
- $p(x \mid c)$ looks strong

But if you intervene:

- Forcing MYC high without changing cell cycle
- Ribosomal genes may barely move

So:

- Observational world says "MYC predicts ribosomes"
- Interventional world says "MYC barely causes ribosomes"

Only the second answer helps drug discovery.

1.5 Conditioning vs Intervention

This is worth saying very cleanly:

- **Conditioning**: "Given that I *observed* c, what do I see?"
- **Intervention**: "Given that I *forced* c, what happens next?"

Conditioning keeps the causal machinery intact. Intervention rips out arrows and replaces them.

In causal graphs (which you'll use later in the project):

- Conditioning filters data
- Intervention **cuts incoming edges** into c

That edge-cutting is why causal reasoning cannot be reduced to statistics alone.

1.6 Why This Matters for Computational Biology

Computational biology-style projects care about questions like:

- “If I inhibit this gene, what happens downstream?”
- “If I target this pathway, does disease state change?”
- “Which transcript is a *lever*, not just a marker?”

Those are all $do(\cdot)$ questions.

Observational ML can tell you:

- Which genes correlate with disease
- Which transcripts are predictive

Causal ML tries to answer:

- Which genes are *drivers*
- Which transcripts are *intervention-relevant*
- Which perturbations will actually move phenotype

That’s why your project structure separates:

- Observational datasets (GTEx, TCGA)
- Perturbation datasets (Perturb-seq, CRISPR screens)

You are literally separating $p(x \mid c)$ data from $p(x \mid do(c))$ data.

1.7 What Comes Next

This distinction sets up everything downstream:

- Why randomized controlled trials work
- Why propensity scores exist
- Why ATE, CATE, and counterfactuals matter
- Why generative models alone are insufficient for causal claims
- Why perturbation biology is gold

Next steps naturally flow into:

- How to estimate $p(x \mid do(c))$ when you *don’t* have interventions
- How biology gives you partial interventions
- How causal graphs formalize this logic

Once this clicks, causal ML stops feeling mystical and starts feeling like *engineering under uncertainty*.