BMS PROJECT

Bristol Myers Squibb®

Computation Tool to Estimate Charge Variants of a Biopharmaceutical Protein

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Presented By:
Pranav S
Glory E
Mushira S

AGENDA

- Charge variants in biopharmaceutical proteins impact product stability and efficacy.
- Proteins have many PTM sites, each adding a small charge shift.
- The total charge distribution becomes extremely complex combinatorially. (Many Combinations)

→ PROBLEM STATEMENT

Need a computation tool that predicts the full charge-variant distribution of a protein by combining the probabilistic charge effects of all its PTMs.

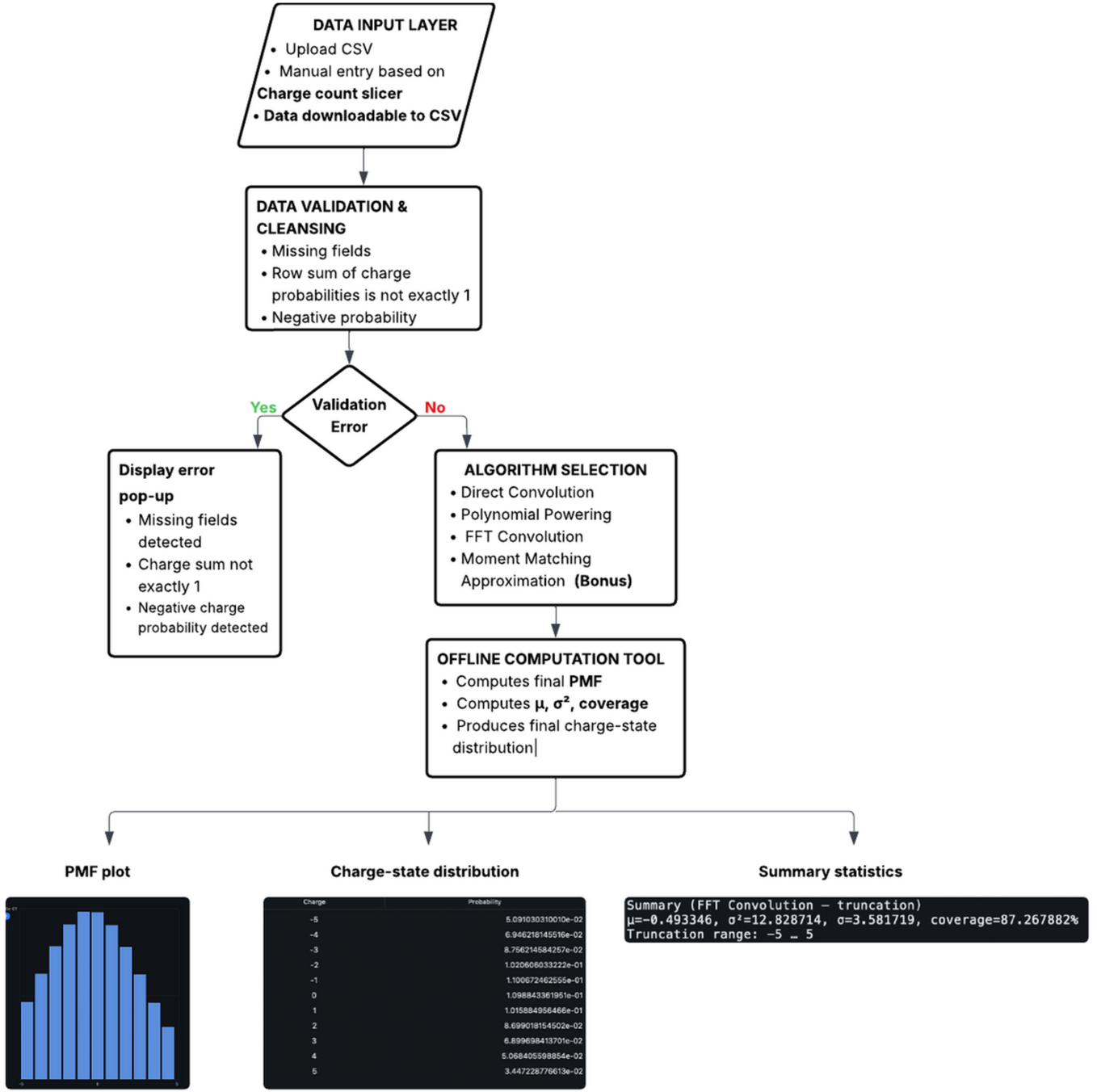
→ OBJECTIVE

- Saves time
- Reduce experimental workload
- Speed up formulation and user-friendly

- CONVOLUTION
- POLYNOMIAL POWERING
- FAST FOURIER TRANSFORM CONVOLUTION
- MOMENT MATCHING APPROXIMATION (BONUS)

→ CORE ALGORITHM IMPLEMENTATION

HIGH-LEVEL BACKEND PIPELINE

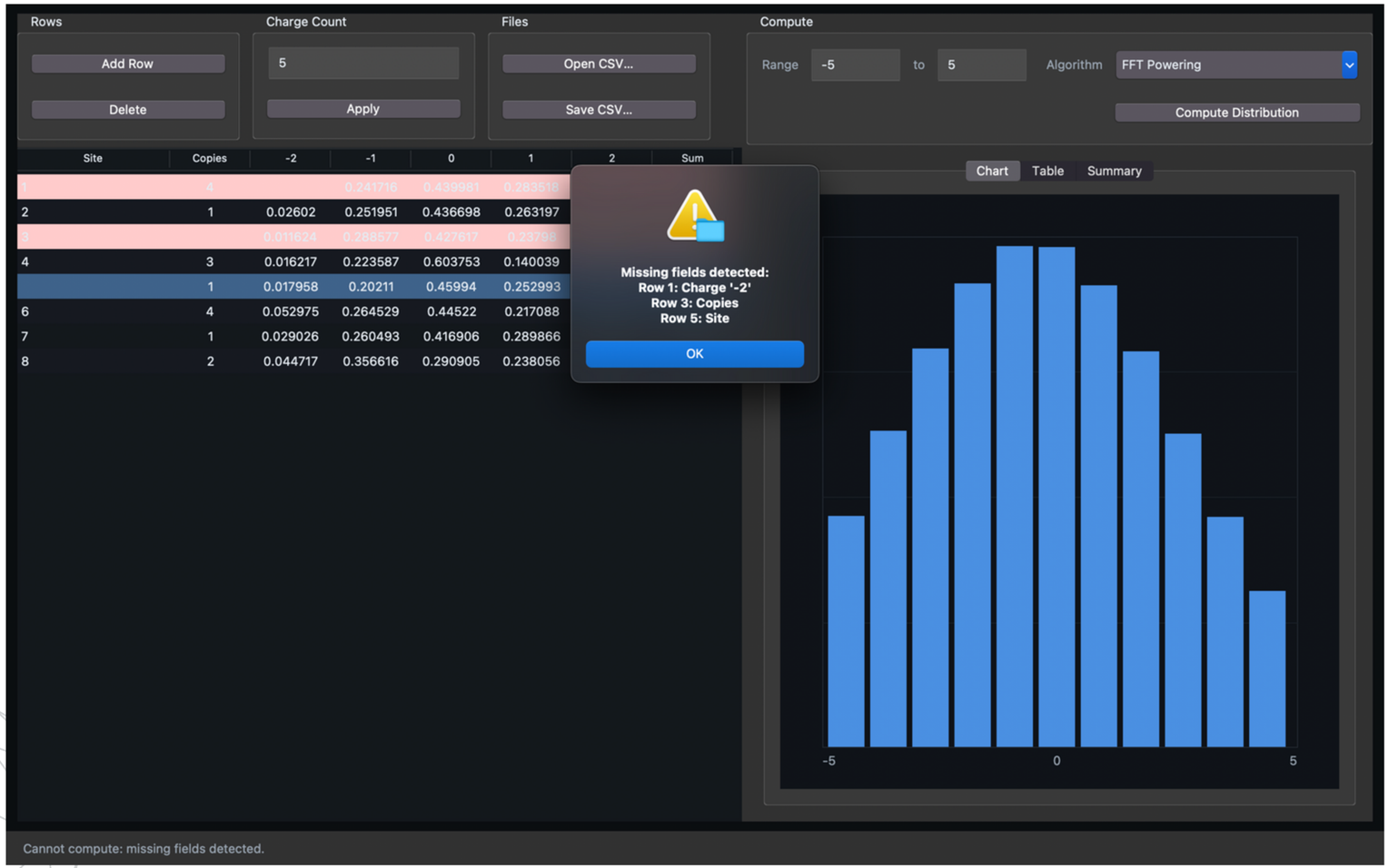


DATA VALIDATION & PREPROCESSING:

DATA CLEANSING

1) Missing Field Checks:

- Detects incomplete **PTM charge probability entries, PTM site** and **copies** and throws an error.

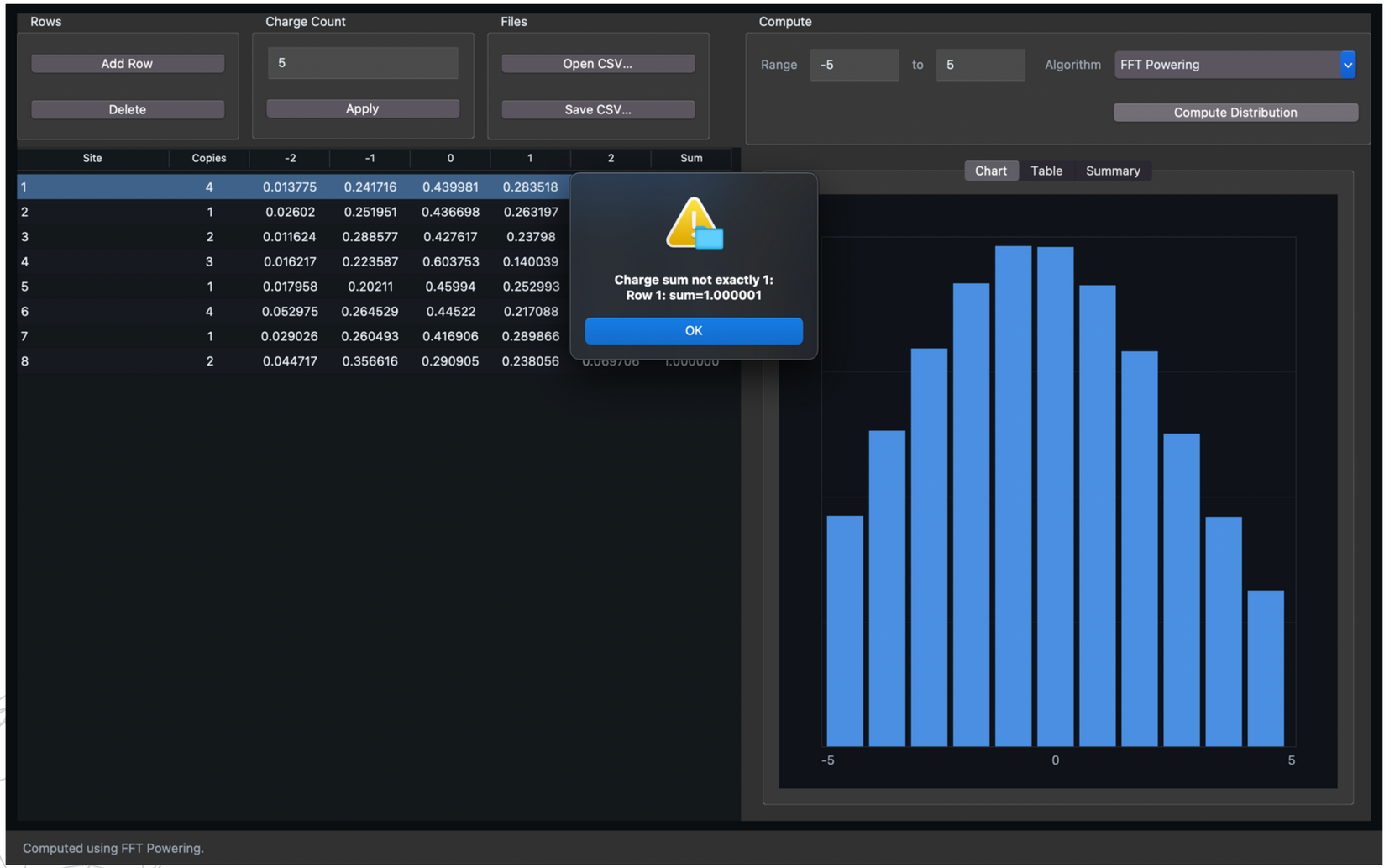


DATA VALIDATION & PREPROCESSING:

DATA CLEANSING

2) Row Sum Enforcement for = 1:

- Displays an error message when **total charge probability in any row is not exactly one.**

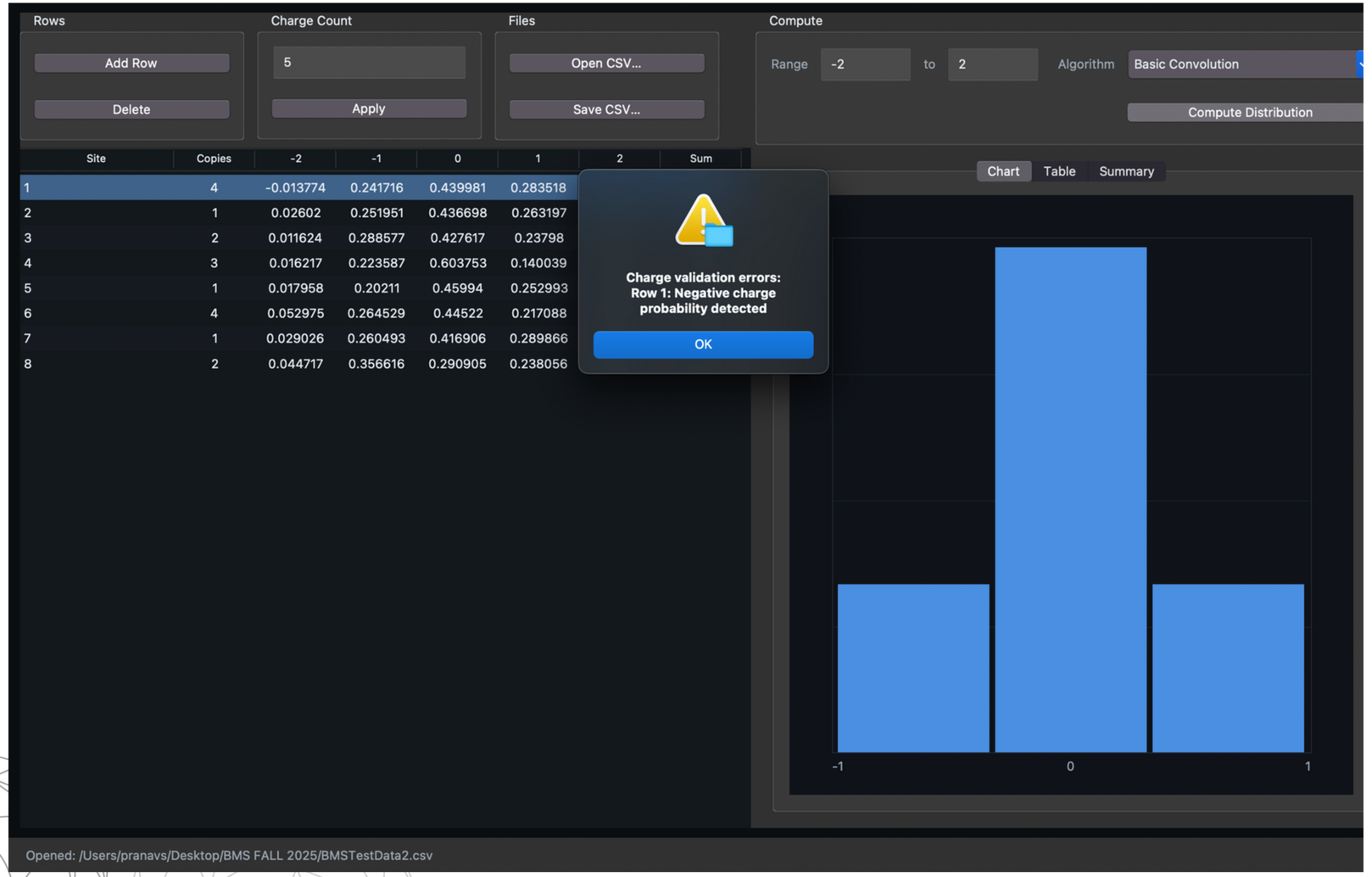


DATA VALIDATION & PREPROCESSING:

DATA CLEANSING

3) Negative probability validation error:

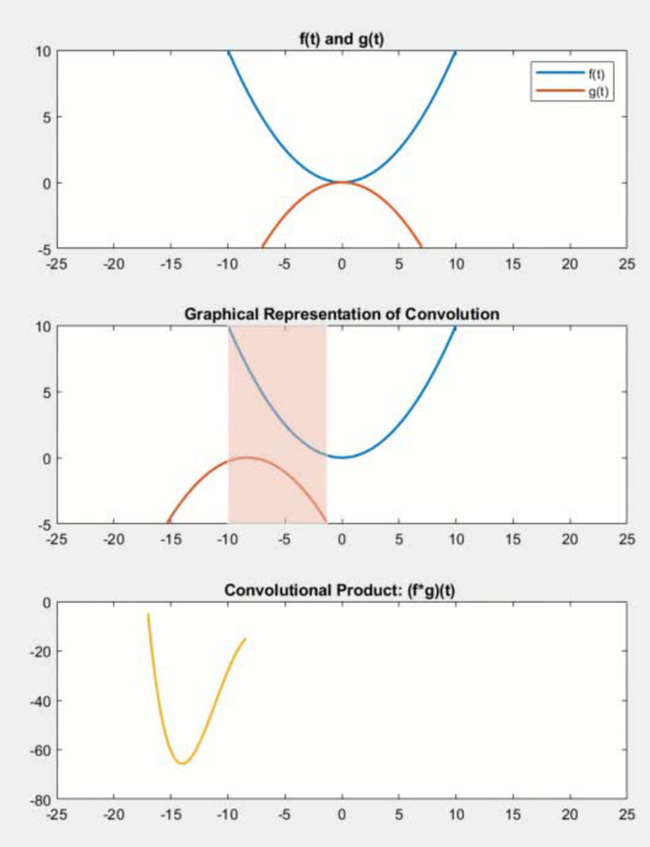
- Flags and displays an error for any negative charge probability in a row.



Convolution - Intuition

In pure mathematical terms, a convolution represents the blending of two functions, f(x) and g(x), as one slides over the other. For each tiny sliding displacement (dx), the corresponding points of the first function f(x) and the mirror image of the second function g(t–x) are multiplied together then added. The result is the convolution of the two functions, represented by the expression [f *g](t).

$$[f * g](t) = \int_0^t f(x)\, g(t-x)\, dx$$



Convolution - Working

Key Idea

- Each copy = Independent discrete random variable over integer charges.
- The total-charge distribution is the discrete convolution of all copy PMFs. (convolve)

Multiple Copies at a Site

- If a site has n identical copies, its site-level PMF is the n-fold self-convolution of the base PMF.
- We compute this with exponentiation-by-squaring a.k.a. binary powering. (power_pmf)
- This reduces the number of convolutions from O(n) to O(log n).

Complexity & Performance

- Let K be the average support size (e.g., ≤ 11 bins for −5…+5).
- Each convolution costs $O(K^2)$; each power_pmf costs $O(K^2 \log n_j)$.
- Memory footprint is O(K) for the PMFs.

$$\text{Time} \approx O\left(K^2 \sum_{j=1}^{S} \log n_j + K^2 S\right)$$

Exact Results, Not Approximations!

Accuracy: coverage% inside the truncation window!

Algorithm 2

Fast Fourier Transform Convolution

Key Idea

- Uses the Convolution Theorem: **convolution in time domain = multiplication in frequency domain.**
- **Converts each PMF (charge distribution) to frequency space** using the Fast Fourier Transform (FFT).
- Performs **pointwise multiplication** of transformed distributions.
- Applies **Inverse FFT** (IFFT) to return to the probability domain.
- Significantly speeds up computation while preserving accuracy.

Multiple Copies at a Site

- For sites with n identical copies, FFT efficiently handles self-convolution of the PMF.
- Instead of n repeated convolutions, the transformation is applied once, and multiplication in frequency space combines all copies.

Complexity & Performance

- Each FFT and IFFT costs **O(N log N**), where N = number of bins (charge states).
- Faster than standard convolution ($O(N^2)$ per step).

BONUS: Moment Matching

Key Idea

- Approximates the total charge distribution using a Normal distribution instead of a full convolution.
- Uses the fact that the sum of many independent charge contributions tends toward a Gaussian **(Central Limit Theorem)**.
  - **Mean:** $\mu = \sum_c c\,p(c)$
  - **Variance:** $\sigma^2 = \sum_c (c-\mu)^2\,p(c)$

Multiple Copies at a Site

If a PTM site has n identical copies, its contribution is simply:

- No repeated convolutions are needed — copies scale the moments linearly.
- Makes the method extremely efficient for large proteins with many repeated PTM sites.
  - $\mu_{\text{total}} += n \times \mu_{\text{site}}$
  - $\sigma^2_{\text{total}} += n \times \sigma^2_{\text{site}}$

Complexity & Performance

- **Moment computation is O(R · C)**
- **n = number of identical PTM copies at a site, W = width (support size) of the charge distribution**
- **N= R x C**
- **Total complexity: O(N + W)**
- Much faster than convolution or FFT-based methods.
- Produces smooth, stable approximations suitable for rapid visualization and exploration.

OFFLINE

- **Offline UI Loading:**
  - The entire user interface is rendered locally, with no online assets, ensuring instant, reliable UI startup.

- **Offline Algorithm Loading:**
  - All computational algorithms run fully on-device, requiring no external services or network connectivity.

- **Offline Resource Loading:**
  - All data, presets, and supporting files are loaded from local storage, eliminating any cloud dependencies.

- **Offline Execution Model:**
  - The tool operates as a fully self-contained executable packaged with all dependencies, enabling true offline functionality.

Tkinter UI is embedded inside the executable, so interface assets (themes, layouts, widgets) load instantly without web resources.

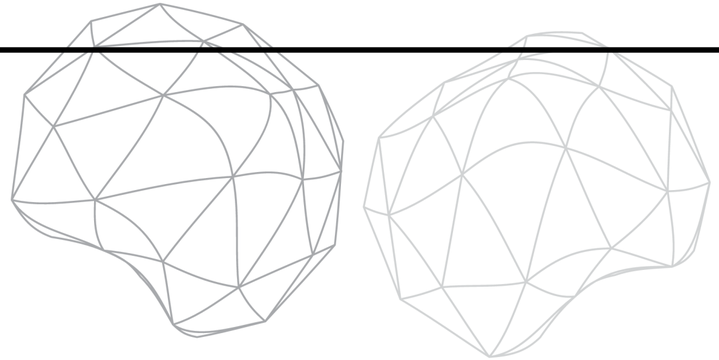
DEMO

COMPARISON:

Algorithm	**Convolution**	**FFT Convolution**	**Moment Matching**
	Exact	Exact	Approximation
Time Complexity	$O(N^2 \log N)$	O(N log N)	**O(N + W)**
Stress Testing	Poor	Poor	Best

FUTURE SCOPE:

- **Enhanced Approximation Models**
  - Add advanced distributions (skew-normal, mixtures, saddlepoint) for more accurate charge predictions.
- **Experimental Data Integration**
  - Allow LC/MS data import to calibrate, validate, and compare predicted vs. observed charge envelopes.
- **Batch & Automation Support**
  - Enable high-throughput processing of multiple constructs and provide a CLI/automation mode.
- **Advanced Visualization & Reporting**
  - Introduce richer plots, overlays, and automated PDF/PowerPoint report generation.
- **Performance & API Expansion**
  - Optimize algorithms with parallelization and expose a Python API for integration into larger analytics pipelines.



Thank You

Any Questions ?