Project Introduction & Vision

- MS data: high-dimensional time–intensity profiles; current ID pipelines are slow and heuristic.
- **Goal**: Create compact vector representations (via JLL + Convex-Hull) that retain peptide-specific features.
- **Benefit**: Faster, more accurate peptide identification in low-dimensional space.
- **Sprint Focus**: Prototype end-to-end pipeline on one test file without PCA, using the convex algorithm.

First Sprint Definition

- **Outcome**:
  1. Ingest one mzML file → build time–intensity matrix (fixed m/z bins, normalize by TIC).
  2. Compute target dimension
  $$k \;=\; \left\lceil \frac{4 \ln n}{\epsilon^2} \right\rceil$$
  (capped by original dimension).
  3. Run JLL (GaussianRandomProjection) → obtain $Y_{\mathrm{JLL}} \in R^{n \times k}$.
  4. Project each row of Y_{JLL} onto its convex hull via constrained optimization.
  5. Generate "fingerprint" vectors for ground-truth peptides (average of their frames' convex embeddings).
  6. Match embeddings to fingerprints (e.g., via cosine similarity) and assign peptides.
- **Backlog**:
  1. **Data Prep**:
     - Parse mzML → intensity matrix over defined m/z bins.
     - Normalize each frame by total ion current.
  2. **Vectorization**:
     - Stack m/z-bin intensities per retention time.
     - (Optional) Apply light smoothing or filter.
  3. **Convex Algorithm & Matching**:
     - Compute k using the JLL lemma.
     - Run GaussianRandomProjection → Y_{JLL}.

- Compute convex hull of Y_{JLL}; project each point onto hull by solving

$$\min_{\alpha} \|V\alpha - y\|^2 \quad \text{subject to} \quad \sum_i \alpha_i = 1,\ \alpha_i \geq 0.$$

- Create reduced-space peptide fingerprints; compute similarities.

4. **Testing & Validation**:
   - Unit tests: check intensity matrix shape/normalization, JLL output dimension, convex-hull projection (points lie on hull, minimal error).
   - Evaluate matching: for each time frame, predicted peptide must match ground-truth within ±0.5min RT.

Test Case & "Line of Truth"

- **Dataset**: Public BSA tryptic digest with known peptide IDs and retention times.
- **Success**: $\geq 70\%$ correct peptide assignments using only convex embeddings.
  - Assignment criterion: time-frame embedding's top-similarity fingerprint must match true peptide RT ±0.5min.
- **Metrics**:
  - Distortion on a sample (e.g., 2 000 frames): mean/max relative squared-distance error.
  - Spearman rank correlation between original and reduced pairwise distances.
  - Visual check: overlay reconstructed (convex) vs. raw intensity profiles for top peptides.

Biggest Uncertainty & Mitigation

- **Uncertainty**: Convex-hull embeddings may blur features of co-eluting peptides.
- **Mitigation**:
  1. Compute distortion metrics (mean distortion < 0.1, rank corr. > 0.8). If unsatisfactory, reduce ϵ ($\rightarrow$ larger k).
  2. Tighten solver tolerances for convex projection; ensure points lie on hull facets.
  3. If needed, insert a shallow autoencoder after JLL before convex-hull projection.
  4. Within 10h: iterate k, solver settings, similarity thresholds—use quick unit tests for feedback.

Agile Alignment & Formal Quality

- **Time-Boxed**: 10h to implement, test, evaluate on one test case.
- **Incremental Delivery**: Data parsing $\rightarrow$ vector build $\rightarrow$ JLL + Convex $\rightarrow$ peptide matching; each step yields a testable artifact.
- **Clear Acceptance**: $\geq 70\%$ peptide assignment accuracy.
- **Rapid Feedback**: Code $\rightarrow$ test distortion & accuracy $\rightarrow$ refine within sprint window.
- **Formatting**: Concise, bulletized structure highlighting problem, sprint scope, algorithm focus (Convex), uncertainties, and agile alignment.

Disclaimer: This text was generated with the help of AI.