Supplementary Notes

Tracking Dataset Reuse in Proteomics: A Comprehensive Analysis of PRIDE Archive Downloads

Suresh Hewapathirana, Jingwen Bai, Chakradhar Bandla, Selvakumar Kamatchinathan, Deepti J Kundu, Nithu Sara John, Boma Brown-Harry, Nandana Madhusoodanan, Marc Riera Duocastella, Juan Antonio Vizcaíno, Yasset Perez-Riverol

Contents

1 S1. Data Processing Pipeline

1.1 Log Processing Workflow

PRIDE download logs are processed through the `nf-downloadstats` Nextflow pipeline (Figure 1), which retrieves raw TSV log files, parses and filters download events, merges records into a consolidated Parquet file, and generates statistics reports. The pipeline produces a 4.7 GB Parquet file containing 159.3 million records optimized for columnar analytics.

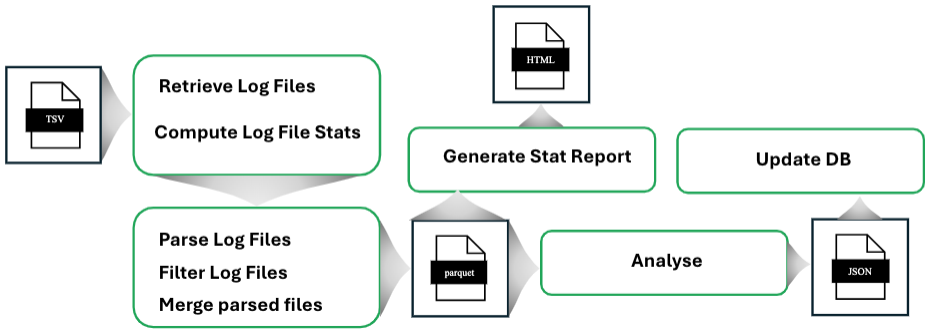

Figure 1: The `nf-downloadstats` workflow for processing PRIDE download logs. Raw TSV logs are parsed, filtered, and merged into a Parquet file for efficient downstream analysis.

1.2 Data Scale and Coverage

The processed dataset covers the period January 2020 through January 2025 (Figure 2). Key metrics include 47.35 million total file downloads across 32,106 distinct projects, 2.26 million unique files, and 807,156 unique users. The analyzed projects represent 96.4% of all public PRIDE datasets, and 88.0% of PRIDE files have been downloaded at least once. Downloads originate from 136 countries with more than 100 downloads each.

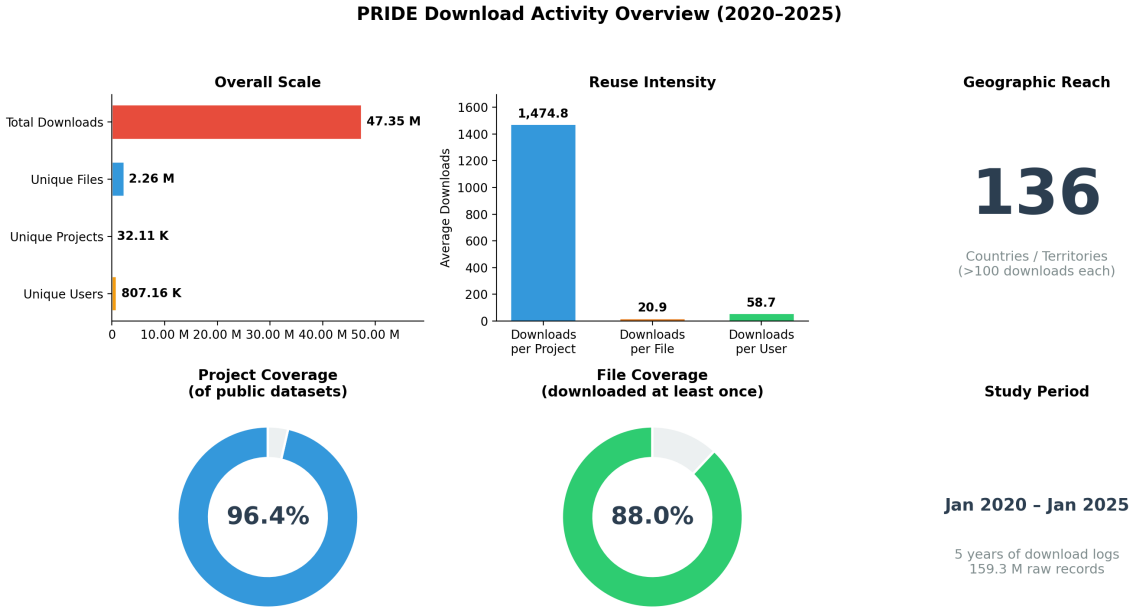

Figure 2: Overview of PRIDE download activity (2020–2025). Overall scale metrics, reuse intensity across projects/files/users, file coverage, and geographic reach.

2 S2. Bot Detection Architecture

2.1 System Overview

DeepLogBot implements a multi-stage bot detection pipeline (Figure 3) that processes download logs through feature extraction, anomaly detection, and hierarchical classification. The system supports two classification methods, each with distinct strengths.

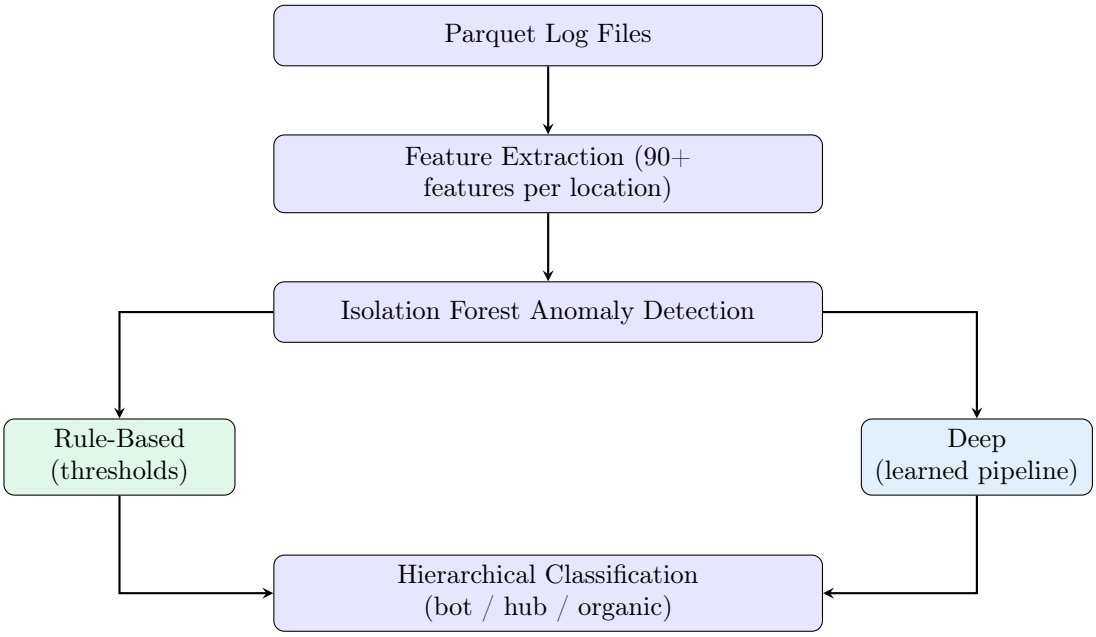

Figure 3: DeepLogBot system architecture. Both methods share feature extraction and Isolation Forest anomaly detection; classification is performed by either a rule-based or learned deep pipeline.

3 S3. Feature Catalog

DeepLogBot computes 90+ features per location, organized into six categories. Table 1 provides the complete catalog.

Table 1: Complete feature catalog organized by category.

Feature	Description
Basic Activity (21 features)	
`unique_users`	Number of distinct user identifiers
`downloads_per_user`	Total downloads / unique users
`total_downloads`	Aggregate download count
`projects_per_user`	Unique datasets / unique users
`avg_users_per_hour`	Average unique users across active hours
`max_users_per_hour`	Peak unique users in any single hour
`user_cv`	Coefficient of variation of users per hour
`users_per_active_hour`	Users per hour (only counting active hours)
`hourly_download_std`	Standard deviation of hourly download counts
`peak_hour_concentration`	Fraction of downloads in the busiest hour
`working_hours_ratio`	Fraction during 9AM–6PM local time
`hourly_entropy`	Shannon entropy of hourly distribution

Feature	Description
night_activity_ratio	Fraction during 10PM–6AM
yearly_entropy	Distribution uniformity across years
peak_year_concentration	Max year's fraction of all downloads
years_span	Number of years with activity
downloads_per_year	Average annual download count
year_over_year_cv	CV of yearly download counts
fraction_latest_year	Latest year's share of total
is_new_location	Binary: only active in latest year
spike_ratio	Latest year / historical average
Advanced Behavioral (15 features)	
burst_pattern_score	Concentration of downloads in short time windows
circadian_rhythm_deviation	Deviation from typical human circadian patterns
user_coordination_score	Synchronized activity across user IDs
hourly_cv_burst	CV of hourly counts (burst detection)
spike_intensity	Magnitude of download spikes
user_peak_ratio	Peak-hour users / average users
night_ratio_advanced	Refined night activity metric
work_ratio_advanced	Refined working hours metric
evening_ratio	6PM–10PM activity fraction
morning_ratio	6AM–9AM activity fraction
user_coordination_std	Variability in coordination patterns
avg_concurrent_users	Average concurrent active users
max_concurrent_users	Peak concurrent users
is_bursty_advanced	Binary: exhibits burst patterns
is_nocturnal	Binary: predominantly night activity
Bot Interaction (6 features)	
dl_user_per_log_users	Downloads per user normalized by log(users)
user_scarcity_score	Inverse user density measure
download_concentration	Gini of download distribution across users
temporal_irregularity	Non-uniformity of temporal access patterns
bot_composite_score	Weighted combination of bot indicators
anomaly_dl_interaction	Anomaly score $\times$ download concentration
Bot Signature (8 features)	
request_velocity	Downloads per active day
access_regularity	Regularity of access intervals
ua_per_user	User agent diversity per user
ip_concentration	IP address concentration
session_anomaly	Deviation from normal session patterns
request_pattern_anomaly	Unusual request sequences
weekend_weekday_imbalance	Weekend vs weekday activity ratio
is_high_velocity	Binary: extremely high request rate
Discriminative (17 features)	
file_exploration_score	Breadth of file access patterns
file_mirroring_score	Consistency with mirroring behavior
file_entropy	Shannon entropy of file access distribution
bot_farm_score	User homogeneity (coordinated fakes)
user_authenticity_score	Diversity of user behavior patterns
user_homogeneity_score	Similarity across users at location

Feature	Description
geographic_stability	IP geographic consistency
version_concentration	Concentration in file versions
targets_latest_only	Binary: only accesses latest versions
unique_versions	Number of distinct file versions
lifespan_days	Duration of activity
activity_density	Active days / total lifespan
persistence_score	Long-term consistent access pattern
malicious_bot_score	Composite malicious indicator
legitimate_automation_score	Composite legitimate indicator
bot_vs_legitimate_score	Differential (malicious − legitimate)
is_likely_malicious	Binary: likely malicious automation
Time Series (23 features)	
Outburst Detection (6 features)	
outburst_count	Number of spikes (> 2 standard deviations)
outburst_intensity	Average spike magnitude
max_outburst_zscore	Highest Z-score across time windows
outburst_ratio	Fraction of activity in outbursts
time_since_last_outburst	Recency of latest spike
longest_outburst_streak	Max consecutive high-activity periods
Periodicity Detection (4 features)	
weekly_autocorr	Autocorrelation at 7-day lag
dominant_period_days	Most significant period (FFT)
periodicity_strength	Strength of dominant period
period_regularity	Consistency of the period
Trend Analysis (5 features)	
trend_slope	Linear trend direction (normalized)
trend_strength	R^2 of linear fit
trend_acceleration	Second derivative
detrended_volatility	Volatility after removing trend
trend_direction	Categorical (-1, 0, $+1$)
Recency-Weighted (4 features)	
recent_activity_ratio	Recent 30 days vs historical average
recent_volatility_ratio	Recent CV vs historical CV
recency_concentration	Fraction in last 30 days
momentum_score	Exponentially-weighted trend
Bot Signature Temporal (3 features)	
autocorrelation_lag1	Day-to-day correlation
circadian_deviation	Distance from human circadian pattern
request_timing_entropy	Entropy of request timing

4 S4. Algorithm Details

4.1 Rule-Based Method

The rule-based method applies threshold patterns from a YAML configuration file. Classification proceeds in two stages to assign each location to one of three categories—**bot**, **hub** (legitimate

automation), or **organic**:

1. **Stage 1 (Organic vs. Automated):** Organic patterns match on `working_hours_ratio` ≥ 0.4 and `regularity_score` ≤ 0.6, or `interval_cv` ≥ 0.7, or `unique_users` < 50 with moderate activity. Automated patterns match on `regularity_score` ≥ 0.7, or `night_activity_ratio` ≥ 0.35 with low working hours, or `user_coordination_score` ≥ 0.6 with many users.

2. **Stage 2 (Bot vs. Hub):** Among automated locations, bot patterns include many-users-low-downloads (`unique_users` ≥ 1000, `downloads_per_user` ≤ 50), coordinated activity (`coordination_score` ≥ 0.7, `authenticity_score` ≤ 0.4), and suspicious timing (`night_activity_ratio` ≥ 0.5, `working_hours_ratio` ≤ 0.2). Hub patterns match on mirror-like behavior (`downloads_per_user` ≥ 500, `unique_users` ≤ 100) or CI/CD patterns (`users` ≤ 10, `regularity` ≥ 0.7).

4.2 Deep Architecture Method

The deep method implements a five-stage learned pipeline that replaces hand-tuned thresholds with data-driven decisions:

1. **Seed Selection.** High-confidence organic, bot, and hub locations are identified from feature distributions (e.g., download-per-user thresholds, working hours ratio). These seeds provide training labels for downstream stages.

2. **Organic VAE.** A variational autoencoder is trained on organic seed locations to learn the manifold of normal download behavior. Locations with high reconstruction error are flagged as anomalous.

3. **Deep Isolation Forest.** Non-linear anomaly detection via neural projections (DeepOD library, with scikit-learn Isolation Forest as fallback) captures complex anomaly patterns in the feature space.

4. **Temporal Consistency.** Modified z-score spike detection without fixed thresholds identifies temporal anomalies by comparing each location's download patterns against robust statistical baselines.

5. **Fusion Meta-Learner.** A gradient-boosted classifier combines all anomaly signals - VAE reconstruction error, deep IF scores, temporal anomaly scores, and 33 behavioral features - into calibrated three-class probabilities (bot/hub/organic) via Platt scaling.

Additionally, soft priors encode pre-filter signals as continuous features (no hard lockout), and a reconciliation step overrides predictions when the pipeline and pre-filter strongly disagree. A post-classification hub protection step prevents legitimate automation from being misclassified as bots.

5 S5. Ground Truth Construction

Ground truth labels were assigned using high-confidence heuristic criteria applied to a 1-million record sample (Table 2):

Table 2: Ground truth label criteria and counts.

Label	Subtype	Count	Criteria
Bot	Ground truth bot	–	≥10K users, ≤10 DL/user
	Large-scale bot	–	≥5K users, ≤25 DL/user
	Bot farm	–	≥1K users, ≤50 DL/user, work ratio ≤0.3
	Total	**88**	
Hub	Mirror	–	≤5 users, ≥1K DL/user
	Institutional hub	–	≤20 users, ≥500 DL/user
	Research hub	–	10–200 users, ≥200 DL/user, ≥100K total, work ratio ≥0.2
	Total	**44**	
Organic	Individual user	–	≤3 users, ≤20 DL/user, work ratio ≥0.4
	Research group	–	3–30 users, 5–100 DL/user, work ratio ≥0.35
	Casual user	–	≤5 users, ≤50 DL/user, night ratio ≤0.3
	Total	**1,279**	
Uncertain	–	18,634	Excluded from benchmark

6 S6. Benchmark Details

6.1 Per-Category Performance

Figure 4 shows precision, recall, and F1 per category for each method. The Rules method achieves the highest bot precision (0.506) but low hub recall (0.159). The Deep method balances all categories, with high hub precision (0.824) and the best hub recall (0.636).

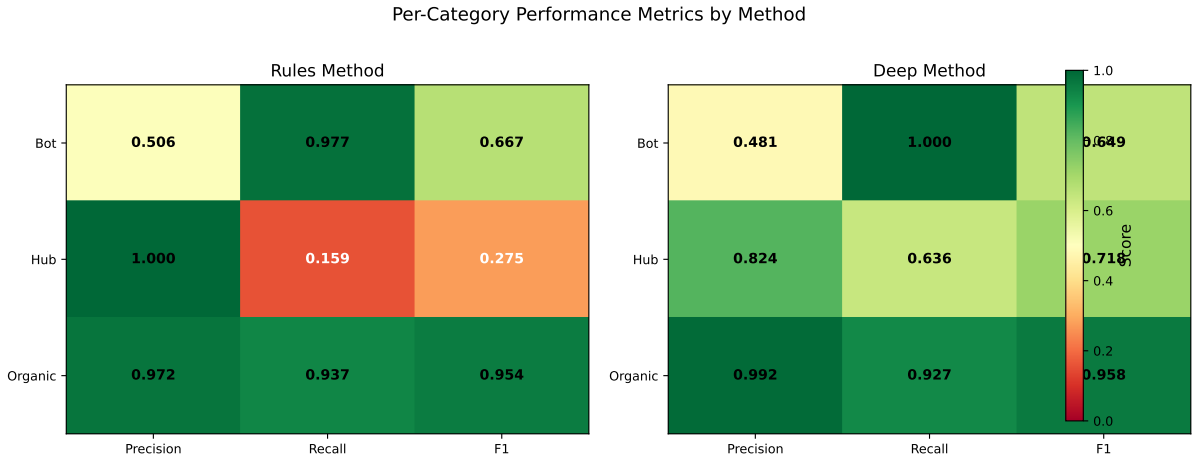

Figure 4: Per-category precision, recall, and F1 scores for each detection method.

6.2 Bootstrap Confidence Intervals

Macro F1 scores with 1,000-iteration bootstrap 95% confidence intervals (Figure 5): Deep achieves 0.775 [0.731, 0.818] and Rules achieves 0.632 [0.574, 0.691]. The non-overlapping CIs confirm a statistically significant performance difference between the two methods.

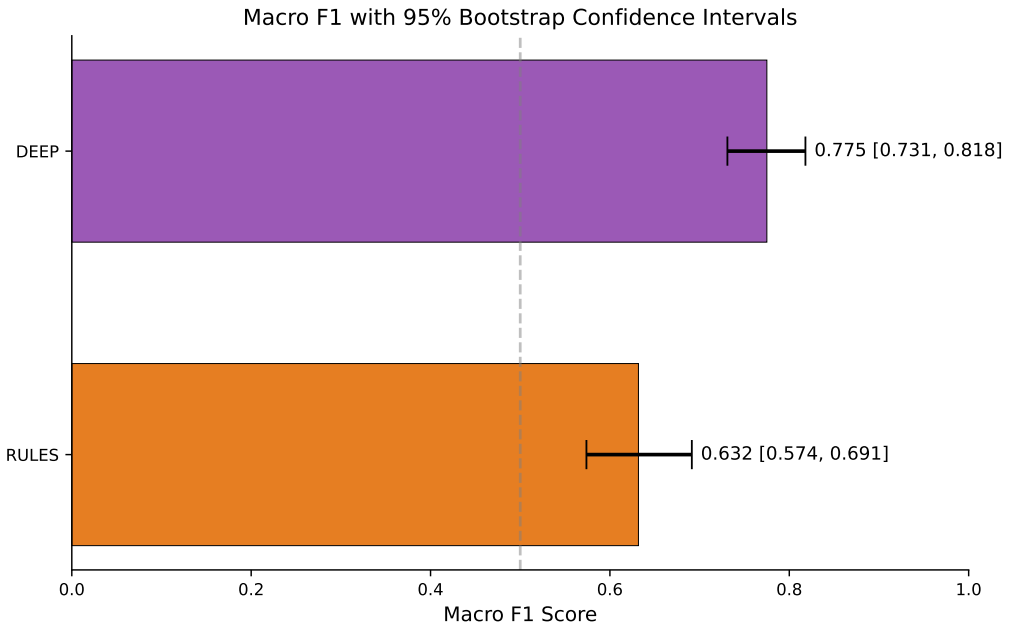

Figure 5: Macro F1 scores with 95% bootstrap confidence intervals for both methods.

6.3 Statistical Significance

McNemar's test with continuity correction was used to compare methods pairwise (Table 3):

Table 3: Pairwise McNemar's test results.

Comparison	χ^2	**p-value**	**Significant?**
Rules vs Deep	0.024	0.877	No

Rules and Deep do not differ significantly ($p = 0.877$).

6.4 Inter-Method Agreement

Figure 6 shows pairwise agreement between methods. Rules and Deep agree on 87.8% of classifications (Cohen's $\kappa = 0.508$), indicating moderate agreement. Per-category agreement is highest for organic classification and lowest for hubs.

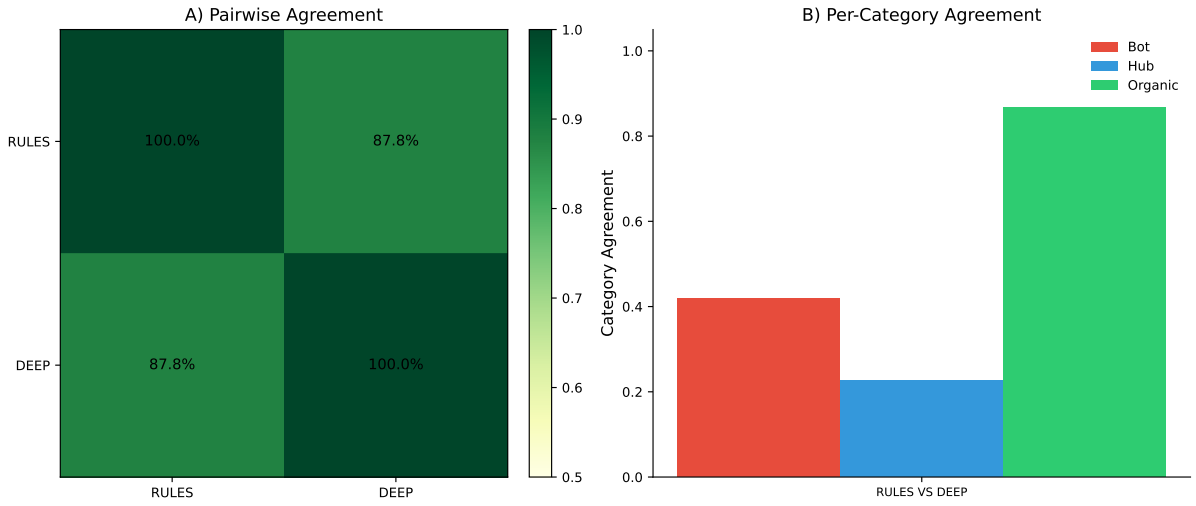

Figure 6: Inter-method agreement analysis. (A) Pairwise overall agreement matrix. (B) Per-category agreement across method pairs.

6.5 Classification Outcome Comparison

On the 1M-record benchmark sample, the two methods produce different classification distributions (Figure 7). The Rules method classifies 29% of locations as bots (72% of downloads), while Deep classifies 34% (77% of downloads).

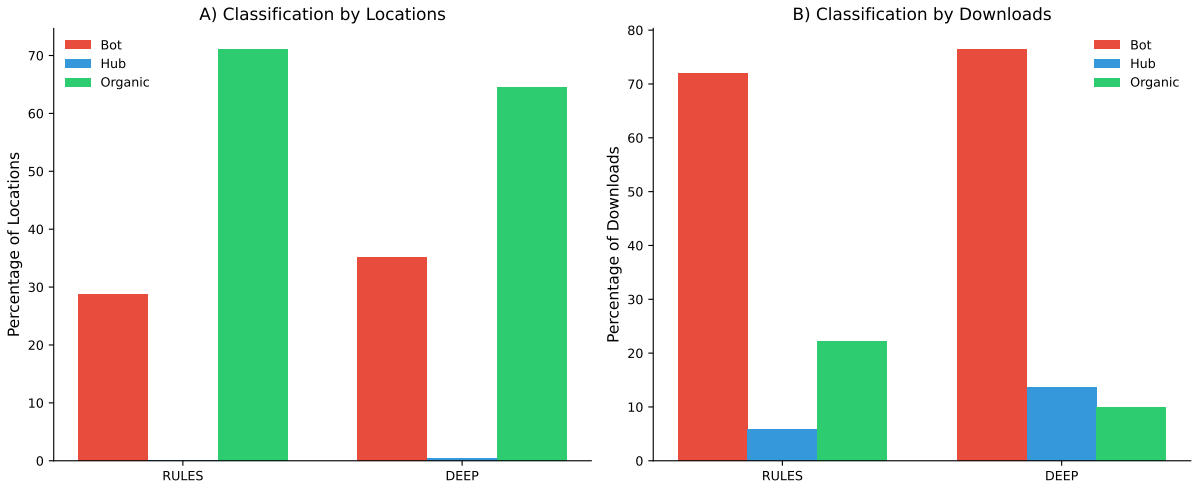

Figure 7: Classification outcome comparison. (A) Percentage of locations classified in each category. (B) Percentage of downloads attributed to each category.

7 S7. Bot Removal Analysis

7.1 Full-Dataset Classification

The Deep method - the best-performing algorithm in our benchmark (macro F1 = 0.775) - was applied to the complete dataset of 71,133 locations aggregated from 159.3 million download records. The classification results are summarized in Figure 8:

- **Bot:** 37,779 locations (53.1%), accounting for 88.0% of downloads
- **Hub:** 664 locations (0.9%), accounting for 11.3% of downloads (institutional mirrors/aggregators)
- **Organic:** 32,690 locations (46.0%), accounting for 0.7% of downloads (genuine individual users)

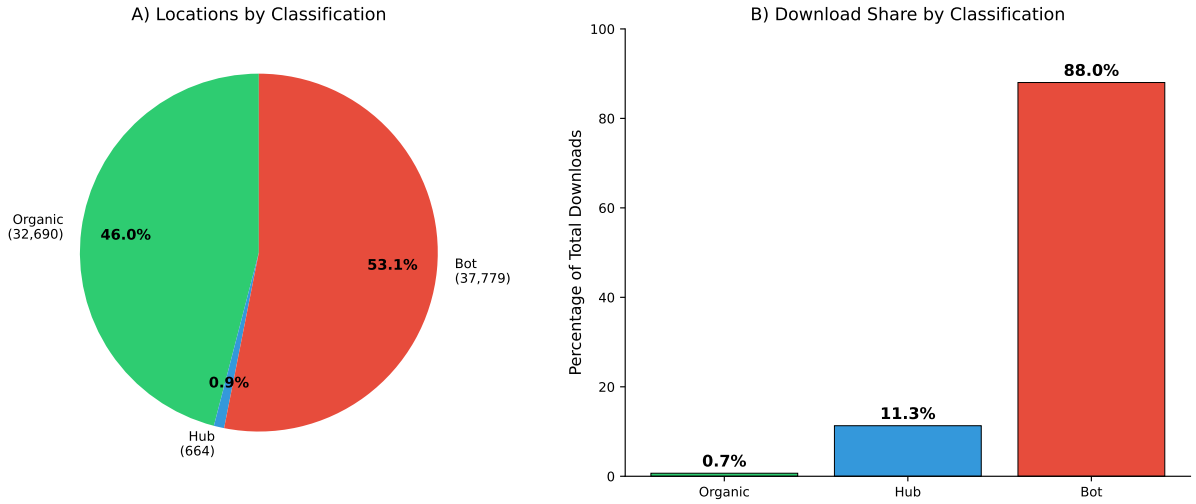

Figure 8: Full-dataset classification results. (A) Location distribution: roughly equal numbers of bot and organic locations. (B) Download distribution: bots generate 88.0% of all traffic despite representing 53.1% of locations.

The asymmetry between location counts and download volumes is striking: bots generate far more traffic per location on average, while institutional hubs - though comprising only 0.9% of locations - account for 11.3% of all download volume.

7.2 Geographic Impact of Bot Removal

Bot removal changes the apparent geographic distribution of PRIDE usage (Figure 9). After removing bot locations, the United States leads with 5.1M downloads (26.8%), followed by the United Kingdom (4.5M, 23.6%) and Germany (4.3M, 22.5%). China, which dominates raw download volume before filtering, drops to sixth position (464K, 2.4%) after bot removal, indicating that the vast majority of traffic from Chinese IP addresses was automated. Similarly, countries like Brazil and Hong Kong show significant decreases after filtering, confirming that much of their raw traffic was bot-generated.

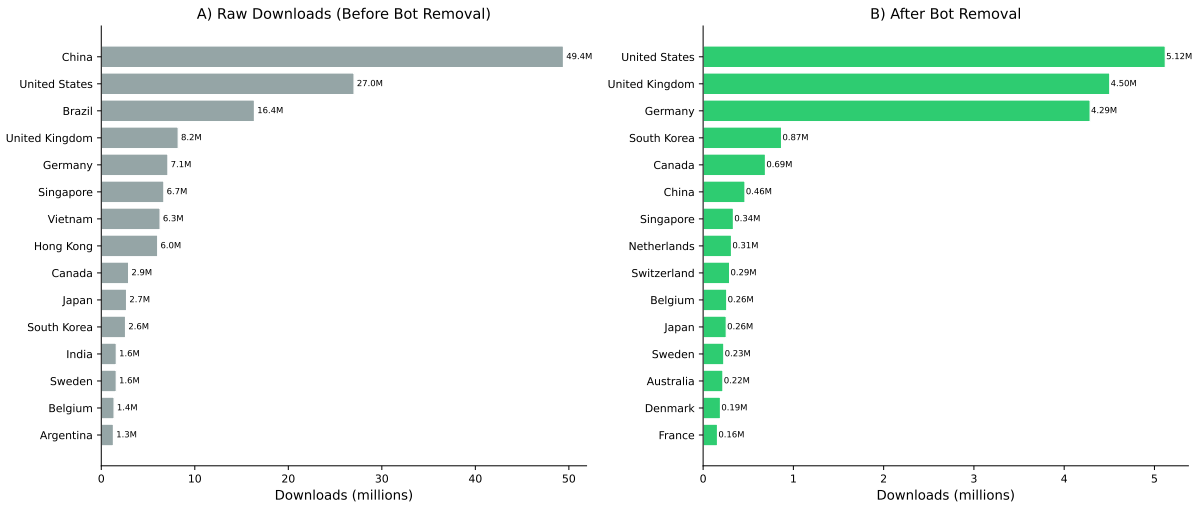

Figure 9: Geographic distribution before and after bot removal. (A) Raw download counts dominated by bot traffic from East Asian locations. (B) After bot removal, the United States and Germany lead genuine usage.

8 S8. Extended Usage Analysis

8.1 Monthly Download Trends

Monthly download patterns (after bot removal) reveal temporal structure within the yearly trends (Figure 10). Activity shows seasonal variation with increased downloads during the academic year and a notable surge in early 2025, likely reflecting both genuine growth and the inclusion of January 2025 data.

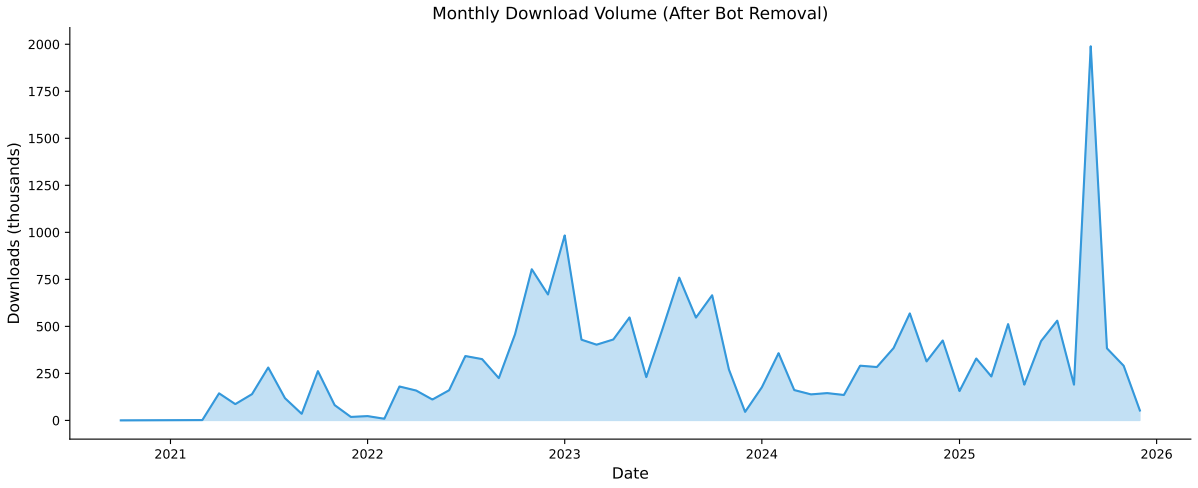

Figure 10: Monthly download volume (after bot removal), showing seasonal patterns and overall growth.

8.2 Hourly Activity Patterns

Hourly download patterns (Figure 11) after bot removal show clear circadian rhythms consistent with human usage: peak activity occurs during daytime hours (UTC), with reduced activity

during nighttime and weekends. This pattern validates the bot removal process, as genuine human users exhibit regular circadian behavior while bots typically show more uniform 24/7 activity.

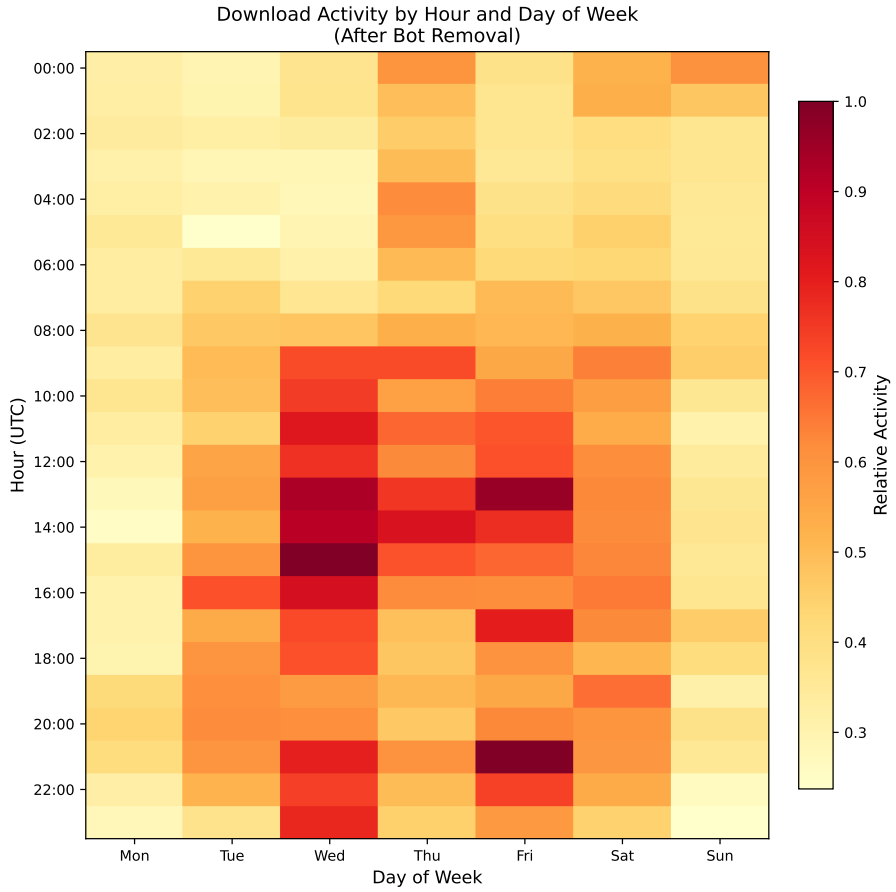

Figure 11: Download activity heatmap by hour (UTC) and day of week, after bot removal. The circadian pattern confirms genuine human usage.

8.3 Regional Distribution

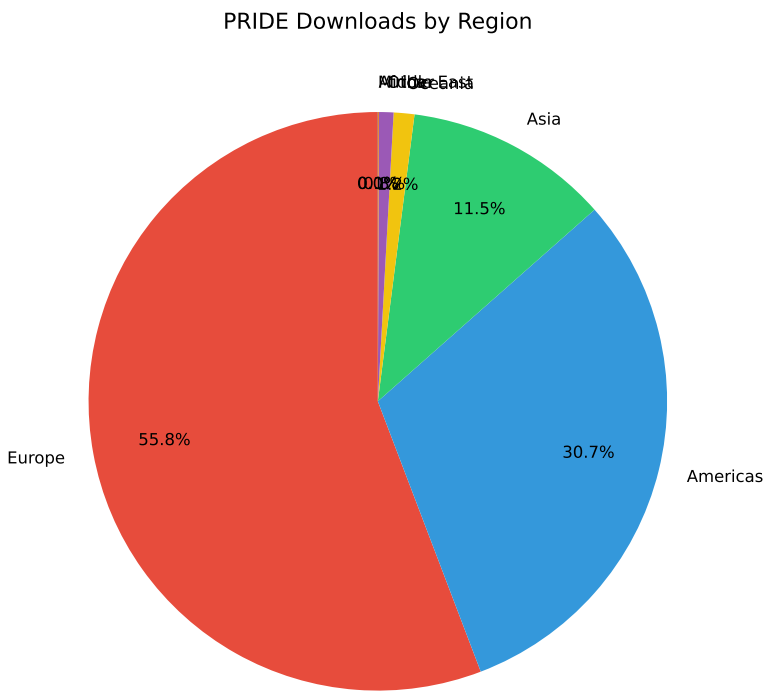

Figure 12: PRIDE downloads by world region (after bot removal, 2020–2025).

8.4 Protocol Distribution

Across the entire study period, FTP accounts for 52.6% of genuine downloads and HTTP for 46.0% (Figure 13). High-performance protocols are emerging: Aspera (FASP) accounts for 0.9% of non-bot downloads and Globus for 0.5%, indicating early but growing institutional adoption of high-throughput transfer tools.

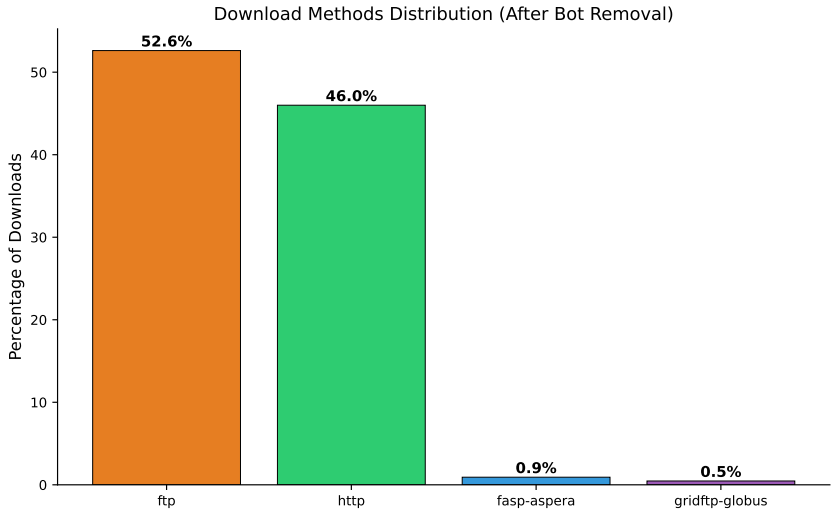

Figure 13: Overall download method distribution across the study period (after bot removal).

8.5 pridepy Adoption

To facilitate adoption of high-performance download protocols, the PRIDE team released `pridepy` [1], a Python command-line tool that abstracts protocol complexity and enables seamless switching between FTP, Aspera, and Globus transfers. The package was published on PyPI in March 2025. Figure 14 shows the monthly download trend throughout 2025, with a total of 6,504 installations. Notably, downloads surged in October 2025 (1,111 installations), coinciding with the rapid growth of Aspera-based transfers observed in the PRIDE download logs (Figure 6B in the main text). This temporal correlation between `pridepy` adoption and Aspera usage growth supports the hypothesis that user-friendly tooling can lower barriers to high-performance protocol adoption in scientific data repositories.

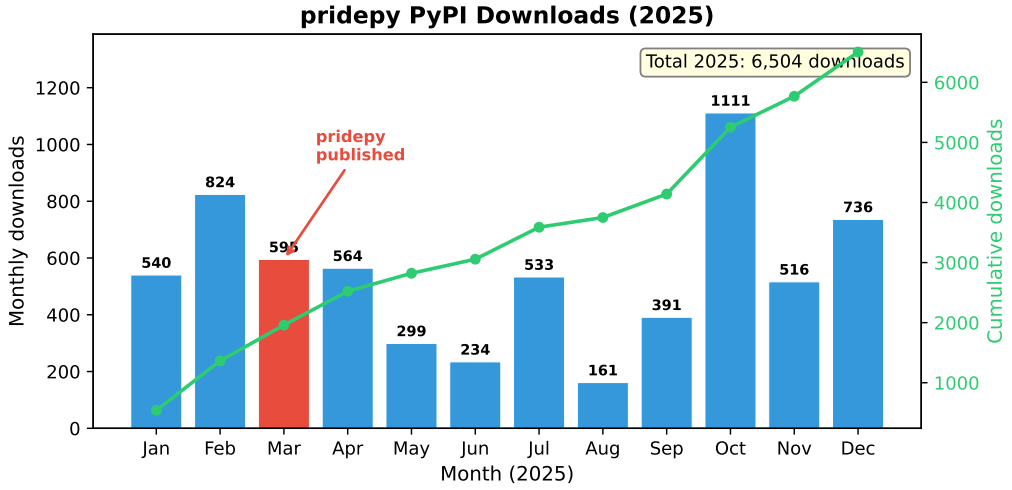

Figure 14: Monthly PyPI download statistics for `pridepy` throughout 2025. The red bar marks the publication month (March 2025). The green line shows cumulative downloads. Data source: pypistats.org (August–December 2025) and ClickPy/ClickHouse (January–July 2025).

8.6 Top Downloaded Datasets

The top 20 most downloaded PRIDE datasets are shown in Figure 15. The most downloaded dataset (PXD004732) has accumulated 355,082 downloads, while PXD017052 reaches the broadest geographic spread (103 countries). Several of the top datasets are large-scale reference proteome studies that serve as benchmarks in the community. Note that dataset rankings depend on the bot detection algorithm applied; the rankings shown here are based on the Deep method.

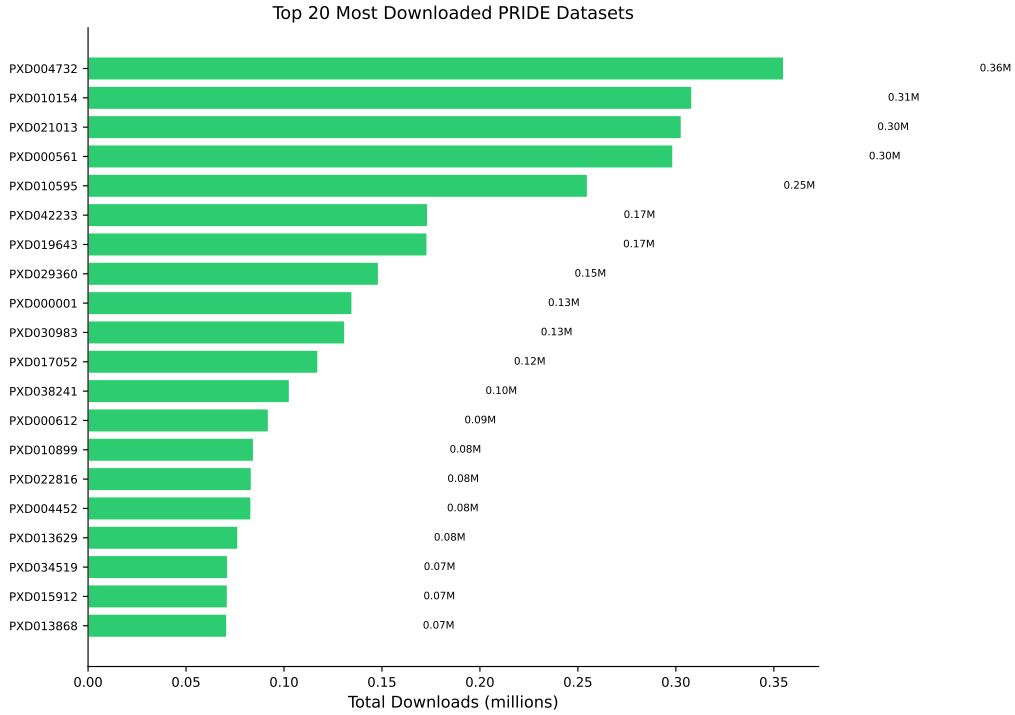

Figure 15: Top 20 most downloaded PRIDE datasets after bot removal, showing download volume and geographic reach.

8.7 Dataset Download Consistency

The consistency heatmap (Figure 7B in the main text) shows that top datasets maintain sustained download activity across multiple years rather than one-time spikes. Beyond this, we rank datasets by a consistency score combining low coefficient of variation with high activity ratio (Figure 16). PXD013868 achieves the highest consistency score (0.788), indicating steady, reliable reuse across the study period.

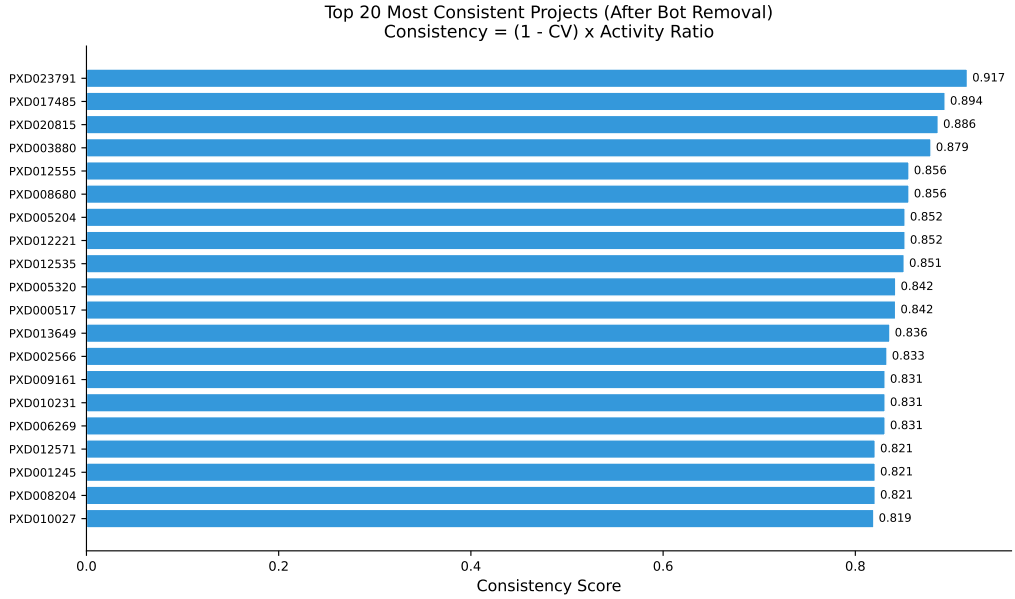

Figure 16: Top 20 most consistently downloaded PRIDE datasets, ranked by consistency score $= (1 - \text{CV}) \times$ activity ratio.

8.8 File-Level Download Distribution

At the individual file level, downloads follow a log-normal distribution (Figure 17), with most files receiving between 3 and 30 downloads. A small number of files exceed 1,000 downloads, representing benchmark datasets and popular reference proteomes.

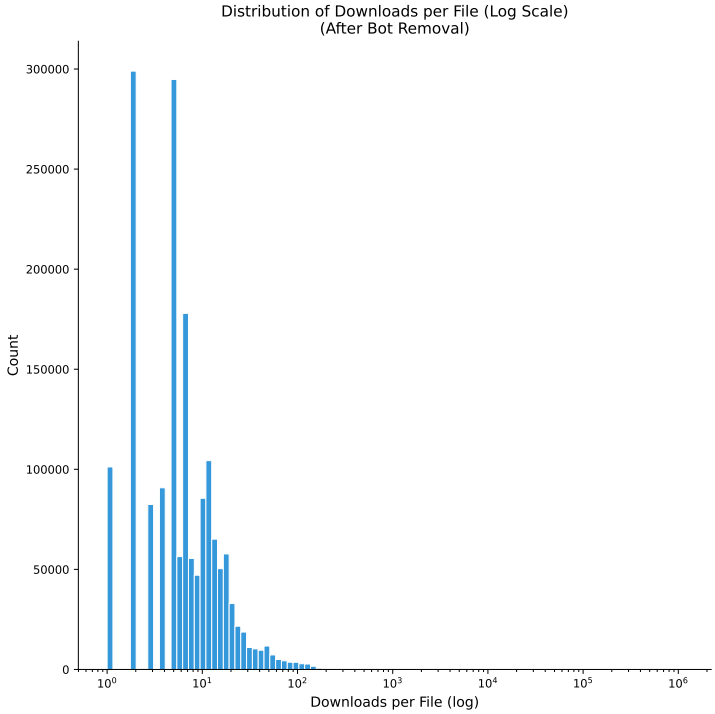

Figure 17: Distribution of downloads per file (log scale). The majority of files receive between 3–30 downloads.

8.9 Country-Level Usage Intensity

The relationship between unique users and total downloads per country reveals distinct usage patterns (Figure 18). Countries cluster along a diagonal indicating proportional usage, while some outliers show disproportionately high downloads per user (suggesting institutional bulk access) or many users with moderate downloads (widespread individual adoption). Bubble size and color represent downloads per user, highlighting intensity differences.

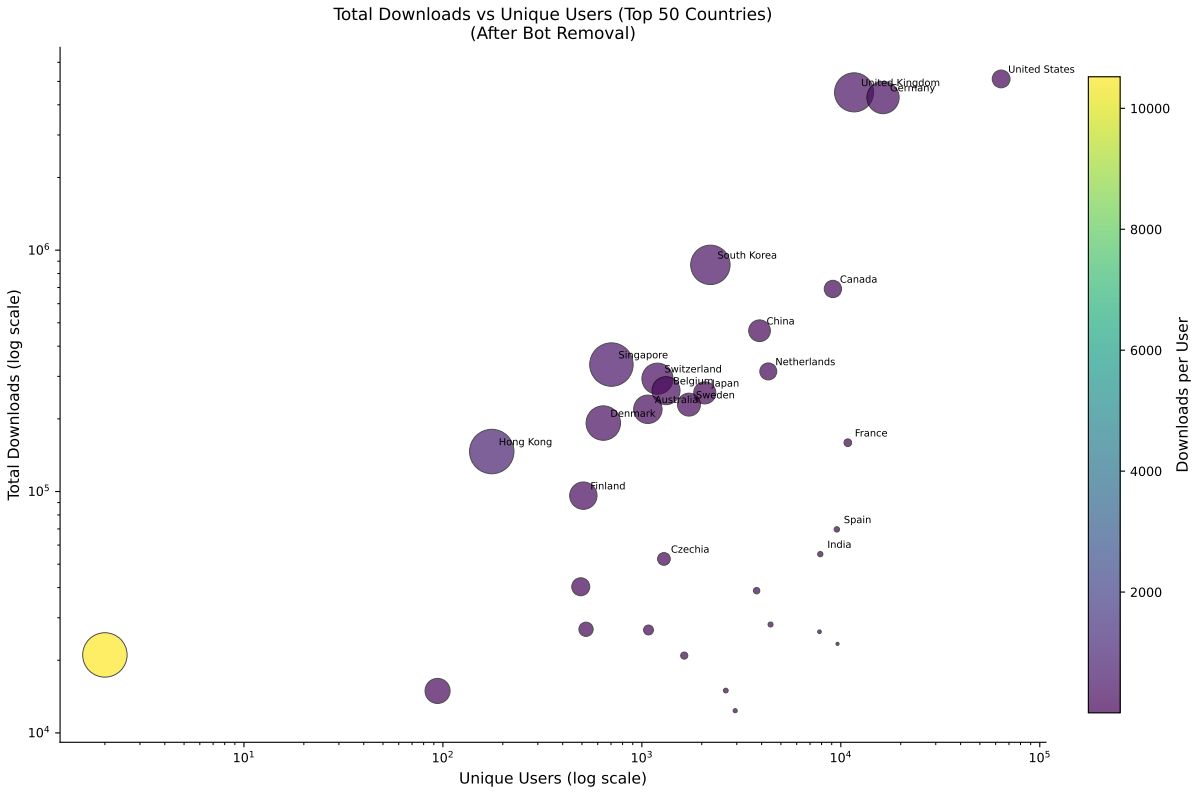

Figure 18: Total downloads vs. unique users per country (log-log scale, after bot removal). Bubble size and color indicate downloads per user. Top 50 countries with >10,000 downloads shown.

8.10 ProteomeXchange Resources

PRIDE hosts 83.2% of all ProteomeXchange datasets, followed by MassIVE (6.9%) and iProX (5.5%) (Figure 19). This dominance reflects PRIDE's position as the primary public repository for mass spectrometry proteomics data.

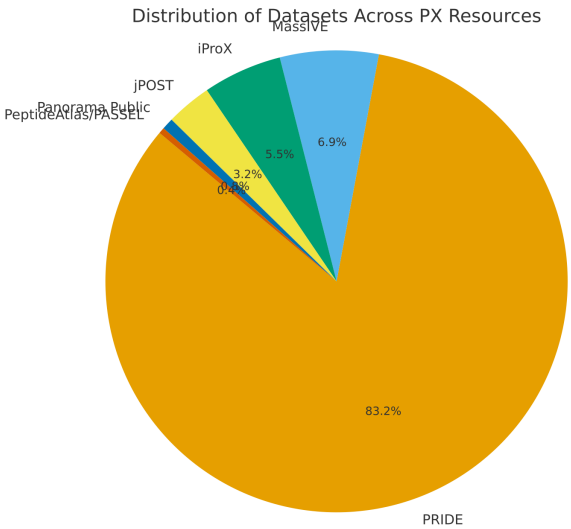

Figure 19: Distribution of datasets across ProteomeXchange partner resources.

9 S9. Limitations

Our ground truth labels are heuristic-derived rather than manually verified, which may introduce systematic biases in the benchmark evaluation. The geographic attribution relies on IP geolocation, which can be inaccurate for users behind VPNs or institutional proxies. The 2025 data is from a partial year, making year-over-year comparisons with full years approximate. Finally, we cannot distinguish multiple individual users who share a geographic location from a single user, which may affect location-level statistics.

References

[1] Selvakumar Kamatchinathan, Suresh Hewapathirana, Chakradhar Bandla, and Yasset Perez-Riverol. pridepy: A Python package to download and search data from PRIDE database. *Journal of Open Source Software*, 10(107):7563, 2025. doi: 10.21105/joss.07563.