

OFFICIAL USER MANUAL

OrdoStat

v 1.0 – PyQt6 Interface

*Calculating Elo performance of chess engines
using maximum likelihood estimation*

AUTHOR

Valter Drazic

SUPPORTED ENGINES

Ordo · BayesElo

LABORATORY

Chroniques Software

EDITION

March 2026



SUMMARY

Table of Contents

01 Introduction & Concept

02 Installation & File
Organization

Recommended structure for the
working directory

03 User Interface — Overview

UI Zones · Language Selection

04 Ordo Mode — Complete
Guide

File Selection · Anchors ·
Launching

05 BayesElo Mode — Complete
Guide

Stdin Sequence · Offset Anchor ·
Results

06 Interpreting Results

Table Columns · Errors \pm ·
Score%

07 Elo Anchoring —
Methodology

Single vs. Multiple Anchors · CSV
File

08 Advanced Functions

Context Menu · Export · Copy
Command

09 Comparison Ordo vs
BayesElo

10 Troubleshooting

Introduction & Concept

OrdoStat is a Python/PyQt6 graphical interface designed to orchestrate two specialized Elo calculation engines for evaluating chess engine performance from tournament PGN files.

Unlike the classic Elo system (incremental, designed for ongoing lists), OrdoStat uses **maximum likelihood** algorithms that calculate the global consistency of all tournament results simultaneously, producing significantly more accurate and robust rankings.

♣ The Problem Solved

You have a PGN file containing the results of a tournament between several chess engines. You want to know their relative — and if possible absolute — Elo strength, calibrated to a known scale.

♣ The OrdoStat Solution

OrdoStat drives Ordo (Miguel Ballicora) or BayesElo (Rémi Coulom), generates a complete ranking with statistical errors, and displays it in an interactive, sortable table.

♣ Mathematical Foundation

For two engines i and j with respective ratings R_i and R_j , the probability of victory follows the logistic function: $\mathbf{P(i,j) = 1 / (1 + 10^{-(R_i - R_j)/400})}$. The algorithms minimize the gap between observed results and these theoretical probabilities.

Installation & File Organization

System Prerequisites

OrdoStat runs on Windows 10/11 (and Linux/macOS provided you have the appropriate binaries). The necessary components are:

COMPONENT	MINIMUM VERSION	ROLE
Python	3.10+	Graphical interface interpreter
PyQt6	6.4+	UI Library (installed via pip)
ordo-win64.exe	1.2.6	Ordo calculation engine (Miguel Ballicora)
bayeselo.exe	0056	BayesElo calculation engine (Rémi Coulom) — optional
OrdoStat.exe	—	Executable compiled by PyInstaller (standalone mode)

Recommended structure of the working directory

All files can be located in the same folder. OrdoStat automatically detects Ordo and BayesElo as soon as the PGN path is selected, if the executables are present in the same directory.

MyTournament\

⚙ ordo-win64.exe ← Ordo engine (required for Ordo mode)

```
⚙️ bayeselo.exe    ← BayesElo engine (required for BayesElo
mode)
⚙️ OrdoStat.exe    ← the graphical interface itself
📄 TotalGames.pgn  ← your games file
📄 anchors.csv     ← Elo anchors (optional)
📄 rating.txt      ← Ordo results (automatically generated)
📄 rating.csv      ← Ordo results in CSV format (generated)
📄 rating_bayeselo.txt ← BayesElo results (automatically
generated)
```



Automatic Executable Detection

As soon as you load a PGN file, OrdoStat scans the same folder and automatically fills the `Ordo exe` and `BayesElo exe` fields if the executables are present there. Similarly, the `Result .txt` field is pre-filled with `rating.txt` in that folder.



Paths with Spaces or Special Characters

Avoid accents and spaces in the name of the working folder. If your PGN path contains spaces, OrdoStat handles this case automatically, but some versions of bayeselo may have difficulties. OrdoStat creates a temporary file with a safe name to bypass this problem.

Installation via PyInstaller (Standalone Mode)

To distribute OrdoStat without requiring Python on the target machine, use the provided `compiler.bat` script:

```
compiler.bat
```

This script installs dependencies, compiles `ordo_gui.py` into `dist\OrdoStat.exe` via PyInstaller, and offers optional UPX compression. The generated executable is standalone — no Python installation is required on the end-user's machine.

User Interface — Overview




The interface is organized into five vertical zones, stacked from top to bottom:



Language Bar

Five languages are available via the flag buttons at the top right: **FR**, **EN**, **ES**, **NL**, **DE**. The change is instantaneous and applies to the entire interface. The active language is remembered between sessions.

Main Buttons

BUTTON	ACTION
 Calculate Elo Performance	Launches the selected engine (Ordo or BayesElo) on the loaded PGN
 Load Existing Result	Loads and displays an existing <code>rating.txt</code> file without re-running the calculation
 Clear	Empties the result table and the console

Ordo Mode — Complete Guide

• ORDO

Ordo (Miguel Ballicora, v1.2.6) is the primary engine of OrdoStat. It uses a convergent *hill climbing* algorithm to estimate relative strengths consistently across the entire pool of games.

4.1 File Fields

FIELD	MANDATORY	DESCRIPTION	EXAMPLE
PGN	YES	Games file in standard PGN format. Can contain thousands of games. [White] and [Black] tags are used to identify engines.	TotalGames.pgn
Ordo exe	YES	Ordo executable. Accepted names: ordo-win64.exe , ordo-win32.exe , ordo.exe , or ordo (Linux).	ordo-win64.exe
Result .txt	YES	Text output file. Created or overwritten at each calculation. A homonymous .csv file is also generated automatically.	rating.txt

4.2 White Advantage Option (-W)

When this box is checked (default), Ordo automatically calculates and corrects for the advantage linked to playing white. This option is **strongly recommended** for any normal time control tournament. It adds the `-W` switch to the Ordo command.



White Correction Result

OrdoStat displays the calculated value at the bottom of the `rating.txt` file, e.g., `White advantage = 58.12` and `Draw rate (equal opponents) = 50.00%`. This information is not shown in the table but remains in the file.

4.3 Elo Anchors Section

Without an anchor, Ordo produces a **relative** ranking where the internal mean is fixed at 0. To obtain **absolute** values comparable to SSDF, CCRL, or your own reference, the ranking must be anchored.

The anchors section consists of 10 rows, each including:


- **Checkbox** — activates/deactivates this anchor row
- **Dropdown menu** — selects the reference engine (populated from the PGN)
- **Elo field** — Elo value to assign to this engine



Minimum Requirement: 2 Active Anchors

Ordo requires at least 2 active anchors to calibrate the Elo scale. With only one anchor, the calculation is rejected and a warning is displayed. Calibration is more precise with numerous anchors well-distributed across the hierarchy.


Loading an anchors.csv file

The  **Load anchors.csv** button allows you to directly import a list of anchors from a CSV file in the format:


```
"Junior 7", 1914  
"Fritz 6", 1824  
"Deep Junior 7", 1823  
"Rybka 2.4 mp 32-bit 8CPU", 2352  
"Glaurung 2.2 JA 8CPU", 2158  
"Wasp 2.00 8CPU", 2315
```

Engines present in the CSV but absent from the PGN are added to the list without causing an error — Ordo will silently ignore them.

Saving Anchors

The  **Save anchors.csv** button exports active anchors to a new CSV file, reusable for future tournaments.

4.4 Loading Engines from PGN

The  **Load engines from PGN** button parses the `[White]` and `[Black]` tags of the PGN and populates the anchor dropdown menus. This operation is also performed **automatically** as soon as the PGN path is selected.

4.5 Launching the Calculation

1

Automatic Verification

OrdoStat verifies that the three mandatory files are defined and exist on disk. A warning is displayed if any is missing.

2

Writing Temporary Anchor File

If anchors are active, an `_anchors_tmp.csv` file is created in the output folder, passed to Ordo via the `-m` switch, and then deleted at the end of the calculation.

3

Asynchronous Execution

Ordo runs in a separate thread. Console output is streamed in real-time to the Console zone. The interface remains responsive during the calculation.

4


Displaying Results

At the end of the calculation, the table is automatically populated from the generated `rating.txt` file. Engines are sorted by descending Elo.

4.6 Generated Command

OrdoStat assembles the following command (example with anchors and white advantage):

```
ordo-win64.exe -W -m anchors_tmp.csv -p TotalGames.pgn -o rating.txt -c rating.csv
```

The  **Copy Command** button places this exact command in the clipboard, allowing manual execution in a command-line interface.

BayesElo Mode — Complete Guide

• BAYESEO

BayesElo (Rémi Coulom, v0056) is a Bayesian Elo calculation program using a different estimation model than Ordo. It produces asymmetric confidence intervals and manages draw rates rigorously.

5.1 Activating BayesElo Mode

In the **Files** section, select the **BayesElo** radio button. The interface modifies immediately:

- The **BayesElo exe** field appears (the Ordo field remains available but unused)
- The **BayesElo Options** panel becomes visible
- The **Elo Anchors** section is grayed out (BayesElo handles anchoring differently)
- The badge

• BAYESEO

confirms the active mode

i Mode Badge Always Visible

The colored badge (green

• ORDO

or blue

• BAYESEO

constantly indicates which calculation engine will be used when clicking *Calculate*.

5.2 Fields Specific to BayesElo

FIELD	MANDATORY	DESCRIPTION
BayesElo exe	YES	BayesElo executable. Accepted names: <code>bayeselo.exe</code> or <code>bayeselo</code> . Automatically detected if present in the PGN folder.
Anchor (Engine)	NO	Dropdown menu populated from the PGN. Select the engine whose Elo you know and wish to use as a scale reference.
Anchor Elo	IF ANCHOR	Elo value to assign to the anchor engine. BayesElo shifts the entire scale so that this engine is positioned at this value.

5.3 BayesElo Calculation Sequence

OrdoStat drives BayesElo interactively via stdin. The exact sequence sent is:

```
readpgn be_tmp_XXXXXX.pgn    ← temporary copy with safe ASCII name
elo                          ← enters EloRating sub-system
mm                           ← Minorization-Maximization algorithm
exactdist                   ← exact distribution (more precise)
offset 2352 Rybka 2.4 mp 32-bit 8CPU ← anchor (if defined)
ratings                     ← extracts ranking
x                           ← exits EloRating
x                           ← exits BayesElo
```



Why mm then exactdist?

`mm` (Minorization-Maximization) performs a fast pre-convergence. `exactdist` then refines the results by calculating the exact probability distribution. Together, both steps yield the most accurate result.

i Temporary File and Path

BayesElo v0056 does not handle quotes or absolute Windows paths in the `readpgn` command. OrdoStat bypasses this by copying the PGN to a temporary name without spaces (`be_tmp_XXXXXX.pgn`) in the same folder, then launching BayesElo from that folder (`cwd`). The temporary file is deleted automatically after calculation.

5.4 Output Result

BayesElo produces an output like:

Rank	Name	Elo	+	-	games	score	oppo.	draws
1	Dragon by Komodo Chess 64-bit	3825	152	127	45	93%	3269	13%
2	Caissa 1.24 POPCNT 8CPU	3788	144	123	45	92%	3254	16%
...								

OrdoStat parses this output and reformats it to the standard Ordo format, allowing display in the usual results table. The result is also saved in the defined `Result .txt` file.

Interpreting Results

6.1 Table Columns

#

Rank in the ranking by descending order of Elo. The top 3 ranks are highlighted in gold.

Engine

Exact name of the engine as it appears in the `[White]` / `[Black]` tags of the PGN.

Elo

Strength calculated by maximum likelihood. Absolute value if anchors are defined, relative otherwise.

± (Error)

Standard error calculated by the formula $\sigma = 400 \cdot \sqrt{(p \cdot (1-p)/n)}$, where p is the score and n the number of games.

Points

Total points scored (win = 1, draw = 0.5, loss = 0).

Games

Total number of games played by this engine in the tournament.

%

Percentage score: $(\text{Points} / \text{Games}) \times 100$. Direct indicator of global performance.

6.2 Sorting the Table

By default, the table is sorted by **descending Elo**. You can click any column header to sort by that value (ascending/descending).

Numerical sorting works correctly for all columns, including the \pm error.

6.3 Reading the \pm Error

The \pm column gives an idea of ranking precision. For an engine with ± 15 , an Elo difference of at least 30 points would be needed for superiority to be statistically significant at 95% confidence.

i Standard Error vs. Ordo -s (Simulations)

The \pm error displayed by OrdoStat is calculated analytically. For more precise errors (especially with multiple anchors), use the Ordo `-s 1000` option via command-line or "Copy Command" to add the parameter manually.

6.4 Reading Example

#	ENGINE	ELO	\pm	POINTS	GAMES	%
1	Dragon by Komodo Chess 64-bit	3121.2	± 8.2	42.0	45	93%
57	Junior 7	1914.1	± 12.7	10.0	45	22%
67	Fritz 6	1824.3	± 13.1	7.5	45	17%

Reading: Dragon (rank 1, 3121 Elo) scored 42 points in 45 games (93%). Its ± 8 error means its true strength is likely between 3113 and 3129 Elo. Junior 7 (rank 57) and Fritz 6 (rank 67) are the low anchors of the ranking.

Elo Anchoring — Methodology

Anchoring is the process by which a *relative* ranking is converted into an *absolute* ranking, comparable to reference lists like SSDF, CCRL, or your own database.

7.1 Single Anchor vs. Multiple Anchors

Single Anchor

A single anchor translates the entire ranking so that this engine is at the specified value. All other Elos are shifted by the same delta. Simple but sensitive to anchor quality.

Multiple Anchors (Recommended)

Ordo performs a **regression** on several reference points, distributing offsets optimally. With 7 well-chosen anchors, the scale is robust even if one anchor is slightly imprecise.

7.2 Choosing Anchors — Best Practices

- Distribute anchors **across the entire Elo range** of the tournament (weak, average, strong)
- Use engines whose SSDF/CCRL values are well-established over several thousand games
- Avoid engines whose behavior varies by hardware (some NPS-dependent engines)
- Minimum recommended: 5-7 uniformly distributed anchors
- Anchors whose official values date back more than 5 years merit verification

✓ **Anchors Example — Valter Drazic Tournament (Ryzen 9 7950X3D)**

Junior 7 (1914) · Fritz 6 (1824) · Deep Junior 7 (1823) · Shredder 10 (1940) · Glaurung 2.2 (2158) · Rybka 2.4 (2352) · Wasp 2.00 (2315). These 7 anchors cover the 1824-2352 range, sufficient to calibrate a tournament ranging from Fritz 5.32 (~1636) to Dragon (~3121).

7.3 BayesElo Anchoring: The `offset` Command

BayesElo does not accept a CSV file of multiple anchors. Anchoring is done via the `offset` command, which shifts the entire scale by an offset to place the selected engine at the desired Elo:

```
offset 2352 Rybka 2.4 mp 32-bit 8CPU
```

This command is automatically inserted by OrdoStat into the stdin sequence when an anchor is selected in the BayesElo Options panel.

Advanced Functions

8.1 Context Menu (Right-click on the Table)

Right-clicking any row in the results table displays a menu with two actions:



Add "EngineName" as Anchor


Automatically creates a new anchor using the engine of the selected row and its calculated Elo as the reference value. Practical for adjusting anchoring after an initial calculation.



Copy Name


Copies the exact engine name to the clipboard. Useful for pasting into the BayesElo anchor field or an external script.

8.2 Copy Command


The  **Copy Command** button places in the clipboard:

- **In Ordo mode:** the full command with all arguments, executable in a command-line interface
- **In BayesElo mode:** the exact stdin script sent to bayeselo, manually reproducible

8.3 Loading an Existing Result

The  **Load Existing Result** button allows you to display a previously generated `rating.txt` file without re-running the calculation. OrdoStat parses the file and displays the full table. If a homonymous `.csv` file exists, it is loaded as well.

8.4 Saving Results

The  **Save Results** button exports the displayed table to:

- **CSV Format:** all columns separated by commas, importable into Excel
- **TXT Format:** table formatted into aligned columns, directly readable

8.5 Settings Persistence

OrdoStat automatically remembers between sessions:

SETTING	SAVED
File paths (PGN, Ordo exe, BayesElo exe, Result)	✓ Yes
Calculator mode (Ordo or BayesElo)	✓ Yes
BayesElo anchor (name + Elo)	✓ Yes
Interface language	✓ Yes
Window size and position	✓ Yes
Ordo anchors (active rows)	✗ No — to be reloaded via CSV file

Comparison Ordo vs BayesElo

Both engines produce Elo rankings by maximum likelihood, but with different philosophies and capabilities. The choice depends on your goal.

CRITERION	ORDO V1.2.6	BAYESELO V0056
Multiple Anchors (CSV file)	✓ Yes, regression	✗ No — single offset
White Advantage Correction	✓ Automatic (-W)	✓ Integrated
Draw Rate Calculation	✓ Automatic (-D)	✓ Integrated
Error Intervals	🕒 Via simulations (-s)	✓ Native asymmetric
Multiple PGN Files	✓ Yes (switch --)	✗ Single file only
"Floating" Anchors (Bayesian)	✓ Yes (-y)	🕒 Native concept
Engines without Wins/Losses	✓ Handled (floor/ceiling)	✓ Handled
Speed (76 engines, 1697 games)	✓ < 1 second	✓ ~2 seconds

Recommended For

CRITERION**ORDO V1.2.6****BAYESEO V0056**Precise absolute
calibrationBayesian statistical
distribution**Practical Recommendation**

For a calibration tournament with multiple anchors and reference to an external list (SSDF, CCRL), use **Ordo**. For a deeper statistical analysis with asymmetric confidence intervals, or as an independent verification of Ordo results, use **BayesElo**. Both results should be very close if the tournament is well-connected.

Troubleshooting

SYMPTOM	PROBABLE CAUSE	SOLUTION
"0 game(s) loaded" in BayesElo console	PGN path with special characters or unreadable file	OrdoStat creates an ASCII temporary file automatically. Check folder permissions.
Empty table after Ordo calculation	<code>rating.txt</code> file not generated, or Ordo returned an error	Check the console for error code. Verify that the PGN contains valid results (not just <code>*</code>).
"Ordo requires at least 2 anchors"	Only one active anchor checked	Check at least two anchor rows, or use an <code>anchors.csv</code> file with multiple engines.
Anchor dropdown menus are empty	PGN not yet loaded, or PGN without valid White/Black tags	Click "⚙ Load engines from PGN" after selecting your PGN.
BayesElo: "Unknown command: offset"	Wrong engine name in anchor field	The name must be identical to what BayesElo loaded (case-sensitive). Copy the name from the results table via right-click → "Copy Name".
Calculated Elo seems unrealistic	Poorly chosen anchors or poorly connected PGN (isolated groups)	Verify that all engines played against common engines. Ordo displays "WARNING" for isolated groups in the console.

SYMPTOM	PROBABLE CAUSE	SOLUTION
Interface is slow during calculation	Calculation in progress on a large PGN	Normal. Calculation runs in a separate thread; interface remains responsive. Wait for completion (status bar shows "Calculation in progress...").
Executable not automatically detected	Non-standard executable name	Use the "..." button to manually navigate to the executable. Automatically detected names: <code>ordo-win64.exe</code> , <code>ordo-win32.exe</code> , <code>ordo.exe</code> , <code>bayeselo.exe</code> .



PGN with "*" Results

Games without a result (`*`) are ignored by Ordo and BayesElo. If your PGN contains only ongoing games, the calculation will produce no results. Ensure result tags `[Result "1-0"]`, `[Result "0-1"]` or `[Result "1/2-1/2"]` are present.