SignalSmith Backtester User Manual

Wendi Ouyang

September 20, 2025

Contents

1 Platform Overview **3**

2 Interface Layout and Workflow **3**

3 Parameter Panel Details **3**

3.1 Strategy Configuration 3
3.2 Universe Filters 4
3.3 Signal Combination Controls 4

4 Technical Indicator Reference **4**

4.1 Relative Strength Index (RSI) 4
4.1.1 Indicator Logic 4
4.1.2 Controls 4
4.2 Moving Average Convergence Divergence (MACD) 5
4.2.1 Indicator Logic 5
4.2.2 Controls 5
4.3 On-Balance Volume (OBV) 5
4.3.1 Indicator Logic 5
4.3.2 Controls 5
4.4 Exponential Moving Average Cross (EMA Cross) 5
4.4.1 Indicator Logic 5
4.4.2 Controls 5
4.5 Stochastic Oscillator 5
4.5.1 Indicator Logic 5
4.5.2 Platform Status 6
4.6 Average Directional Index (ADX) 6
4.6.1 Indicator Logic 6
4.6.2 Platform Status 6
4.7 Aroon Indicator 6
4.7.1 Indicator Logic 6
4.7.2 Platform Status 6
4.8 Additional Indicators 6

5 Signal Pipeline and PnL Computation **6**

5.1 How Configuration Becomes Signals 6
5.2 PnL and Equity Curve Construction 7

6 Project Architecture and File Responsibilities **7**
6.1 High-Level Components 7
6.2 End-to-End Flow 8
6.3 Extensibility Considerations 8

7 Interpreting Backtest Output **9**
7.1 Aggregated Results 9
7.2 Equity Curve and Drawdown 9
7.3 Signals & Price 9
7.4 Performance Metrics 9
7.5 Trades 9

8 Exporting Results **9**

9 Best Practices and Considerations **9**

10 Appendix: API Field Reference **10**

1 Platform Overview

SignalSmith Backtester is a browser-based analytics workstation built for quant researchers and students who want rapid, indicator-driven equity backtests without writing code. Configure multi-factor signal recipes, slice the universe with rich filters, and instantly inspect equity, drawdown, signals, trades, and performance metrics—complete with CSV and JSON exports for deeper research workflows.

2 Interface Layout and Workflow

The application is divided into two primary areas:

- **Sidebar Form**: Configure the strategy template, date range, technical indicators, and universe filters.
- **Results Panel**: Inspect equity, drawdown, signal, performance, and trade outputs. Each section supports one-click export.

A typical session follows these steps:

1. Select a strategy template, test window, and capital allocation in the **Strategy** card.
2. Enable indicators and tune their parameters in the **Indicators** card.
3. Narrow the security universe and choose a signal-combination policy using the **Universe Filters** and **Signals** cards.
4. Press **Run Backtest**. When processing completes, review the analytics and export CSV or JSON snapshots for further analysis.

3 Parameter Panel Details

3.1 Strategy Configuration

Strategy Choose from `mean_reversion`, `momentum`, or `multifactor`. The selection determines how the engine interprets indicator signals and manages positions.

Backtest Range Date picker defining the start and end of the simulation. The default window covers the trailing year.

Initial Capital Starting portfolio value in USD. Equity curves and position sizes scale to this amount.

Fee (bps) Per-trade transaction cost measured in basis points (1 bps = 0.01%).

3.2 Universe Filters

Sector Multi-select list sourced from the metadata table or a default sector set. Use it to restrict industries.

Market Cap Min/Max Optional USD thresholds for minimum and maximum market capitalisation.

Exclude Tickers Tag input for removing specific symbols; entries are automatically uppercased.

3.3 Signal Combination Controls

Combination Policy Determines how indicator signals aggregate: `any` (triggered if any indicator fires), `all` (require every indicator), or `atleast_k` (require at least k indicators).

k Only used with the "At least k" policy. Specifies the minimum number of indicators that must agree.

Max Horizon (days) Caps the validity window for each signal. Expired signals are discarded.

Histogram Horizon (days) Number of days included when computing histogram statistics for diagnostics.

4 Technical Indicator Reference

Descriptions draw on Investopedia's "Top 7 Technical Analysis Tools"[1] and align with the platform implementation.

4.1 Relative Strength Index (RSI)

4.1.1 Indicator Logic

RSI compares average gains and losses over a lookback window and oscillates between 0 and 100. Values above 70 often signal overbought conditions, while readings below 30 highlight oversold momentum.

4.1.2 Controls

- **RSI Lookback**: Number of trading sessions used in the calculation (n). Shorter windows react faster but introduce more noise.
- **RSI Mode**: Choose `oversold` to buy when RSI falls below the threshold or `overbought` to fade rallies when RSI rises above the threshold.
- **RSI Threshold Slider**: Adjust the trigger level associated with the chosen mode.

[1] Investopedia, "Top 7 Technical Analysis Tools", accessed on September 20, 2025.

4.2 Moving Average Convergence Divergence (MACD)

4.2.1 Indicator Logic

MACD measures the spread between fast and slow exponential moving averages (EMAs). Crossovers or positive territory relative to a signal line highlight potential trend shifts.

4.2.2 Controls

- **Enable MACD**: Toggle inclusion in the signal stack.
- **Fast / Slow**: EMA spans for the fast and slow lines (defaults 12 and 26).
- **Signal**: EMA span for the signal line smoothing the MACD series (default 9).
- **MACD Rule**: Choose `signal` for signal-line crossovers or `positive` to act when MACD rises above zero.

4.3 On-Balance Volume (OBV)

4.3.1 Indicator Logic

OBV accumulates volume by adding it on up days and subtracting it on down days to confirm trend strength. Rising OBV suggests buying pressure supporting price advances.

4.3.2 Controls

- **Enable OBV**: Loads volume data and includes OBV in signal generation.
- **OBV Rule**: `rise` requires OBV to break above its 20-day moving average; `positive` triggers when OBV itself is above zero.

4.4 Exponential Moving Average Cross (EMA Cross)

4.4.1 Indicator Logic

The EMA cross checks whether a short-term EMA overtakes a long-term EMA, producing the classic “golden cross” signal for trend reversals.

4.4.2 Controls

- **Enable EMA Cross**: Toggle participation in the composite signal.
- **Short / Long**: EMA spans governing responsiveness of the short- and long-term averages.

4.5 Stochastic Oscillator

4.5.1 Indicator Logic

The stochastic oscillator gauges where the latest close sits within the recent high-low range, surfacing overbought or oversold extremes.

4.5.2 Platform Status

Backend parameters (`stoch_k`, `stoch_d`, `stoch_rule`) exist but are not yet exposed in the UI. Advanced users can configure them via direct API calls.

4.6 Average Directional Index (ADX)

4.6.1 Indicator Logic

ADX measures trend strength independent of direction. Values above 20 typically indicate a meaningful trend.

4.6.2 Platform Status

The backend supports `adx_n` and `adx_min` for filtering weak trends. UI controls will be added in future releases.

4.7 Aroon Indicator

4.7.1 Indicator Logic

Aroon tracks the number of periods since the last high or low to evaluate bullish and bearish momentum. Aroon Up above 70 with Aroon Down below 30 points to bullish conditions.

4.7.2 Platform Status

Parameters `aroon_n`, `aroon_up`, and `aroon_dn` are available through the API but are currently hidden in the sidebar.

4.8 Additional Indicators

Investopedia also highlights tools such as Bollinger Bands. These are not yet part of the platform but can be considered for future expansions.

5 Signal Pipeline and PnL Computation

5.1 How Configuration Becomes Signals

1. **Form submission**: The sidebar form serialises every selection—strategy, dates, indicator parameters, universe filters, and signal policy—into the request payload sent to the FastAPI endpoint `/run_backtest`.

2. **Backend normalisation**: Helper functions `_map_indicators` and `_build_config` translate generic payload fields into the concrete indicator configuration expected by the backtesting engine (enabling or disabling RSI, MACD, OBV, EMA, etc., and supplying the tuned thresholds).

3. **Per-symbol evaluation**: `run_backtest_for_all` iterates over each ticker in the universe, aligns price/volume series, and executes the indicator-specific calculators such as `calculate_rsi`, `calculate_macd`, `calculate_obv`, and `calculate_ema_cross`.

Each calculator returns both raw indicator values and a boolean series of candidate buy signals based on the parameters supplied.

4. **Signal combination**: The boolean outputs are stacked into a DataFrame and sent through `combine_signals`. Depending on the policy (`any`, `all`, or `atleast_k`) and the `k` value, only dates satisfying the aggregation rule remain marked as tradeable opportunities.

5. **Forward return tagging**: For every triggered date, `compute_forward_returns` produces log returns for horizons 1 through `max_horizon`. Those forward returns, along with metadata (price, ticker, which indicators fired), are stored in the `picks` table that drives the rest of the analytics.

5.2 PnL and Equity Curve Construction

- **Trade approximation**: `_make_trades` treats each pick as entering a long position at the adjusted close on the signal date. Using the one-day forward return (`fwd_ret_1d`) it estimates the next day exit price via `exit_price = enter_price * (1 + ret_1d)`. Profit and loss (PnL) is the difference between exit and entry, and the reported return equals `ret_1d`. Transaction fees are currently not deducted.

- **Equity aggregation**: `_build_equity` groups trades by date, averages their 1-day returns, and cumulatively multiplies `(1 + return)` to derive the equity curve normalised to 1.0 at inception.

- **Drawdown and metrics**: `_compute_drawdown` compares equity to its running maximum to quantify underwater periods, while `_compute_metrics` extracts average daily return, volatility, annualised metrics, Sharpe ratio, maximum drawdown, and ending equity for reporting.

- **Outputs delivered**: The FastAPI layer serialises equity, drawdown, signals, trades, metrics, and universe statistics into the `BacktestResponse` consumed by the front end.

6 Project Architecture and File Responsibilities

6.1 High-Level Components

- **frontend/**: React + Ant Design single-page application that renders the form, charts, and tables. Notable files include `src/App.tsx` (results layout and download handlers), `src/components/SidebarForm.tsx` (parameter form and user guidance), and reusable chart/table components under `src/components/`.

- **backend/**: FastAPI adapter that wraps the legacy backtesting engine. `api_server.py` exposes REST endpoints, handles payload validation, loads market data, translates indicator settings, runs the engine, and formats responses.

- **backtest_system.py**: Core Python library containing indicator calculations, signal combination logic, forward return computation, equity analytics, and the orchestrating function `run_backtest_for_all`.

- **data/**: Expected location for market-wide wide-format price/volume tables (Feather files). The backend reads them to supply historical inputs; the directory is commonly excluded from version control.
- **docs/**: Project documentation, including this manual (`backtesting_manual.tex`) and the GitHub deployment guide (`github_deployment.tex`).
- **Root scripts**: Files such as `backtest_system.py`, `data_pipeline.py`, and `web_ui.py` support data ingestion, experimentation, or alternative interfaces outside the web front end.

6.2 End-to-End Flow

1. **User interaction**: Operators adjust strategy parameters in the React sidebar; local storage allows presets.
2. **API invocation**: Submitting the form triggers `/run_backtest` with a JSON payload capturing indicator configuration, date range, filters, and capital assumptions.
3. **Data loading**: The backend retrieves preprocessed wide tables from `data/`, applies optional sector or ticker filters, and constructs the working universe.
4. **Indicator evaluation**: The backtesting engine computes per-ticker signals according to enabled indicators and user-defined thresholds, combines them, and measures forward returns.
5. **Result packaging**: Equity curve, drawdown, statistics, signal list, and trade approximations are summarised into a single response.
6. **Visualisation**: The front end renders Chart.js/Ant Design components using the response and exposes CSV/JSON download helpers for post-processing.

6.3 Extensibility Considerations

- **Adding indicators**: Implement new calculators in `backtest_system.py`, expose configuration switches in the backend `_build_config`, and extend the React form with explanatory text and inputs.
- **Strategy templates**: The `strategy` field currently labels the run; more advanced behaviour (e.g., multi-day holding logic) can be wired by branching inside the backend or engine based on this field.
- **Cost modelling**: Integrate `fee_bps` by adjusting `_make_trades` to deduct proportional costs and modify equity aggregation accordingly.
- **Data sourcing**: Replace or augment Feather tables via `data_pipeline.py`, ensuring column conventions (date plus ticker columns) align with expectations.

7 Interpreting Backtest Output

7.1 Aggregated Results

The overview card lists the number of securities evaluated (Universe size) and the total trades generated, offering a quick sense of signal density.

7.2 Equity Curve and Drawdown

- **Equity Curve**: Cumulative equity indexed to the starting capital. Export the series via `equity_curve.csv` for custom charting.
- **Drawdown**: Percentage drop from the running equity peak, highlighting peak-to-trough risk.

7.3 Signals & Price

Displays the price series overlaid with generated signals to validate entries against market action. If the backend does not return prices, the chart falls back to the equity curve.

7.4 Performance Metrics

Reports daily average return, daily volatility, annualised return and volatility, Sharpe ratio, and maximum drawdown. Use the CSV export to feed downstream analytics.

7.5 Trades

Summarises each position with entry and exit timestamps, prices, profit and loss, and symbol identifiers for manual review.

8 Exporting Results

Every core output supports download:

- **Download JSON**: Captures the full request payload and response for reproducibility.
- **Download CSV**: Individual exports for equity, drawdown, signals, metrics, and trades.

9 Best Practices and Considerations

1. Tune test windows and parameters cautiously to avoid overfitting historical data.
2. When using `all` or `atleast_k` policies, verify that sufficient signals remain to trade.
3. Volume-dependent indicators (OBV, ADX, etc.) require complete volume data; missing inputs can suppress signals.
4. Archive exported CSVs alongside parameter snapshots to guarantee reproducibility.

5. Advanced users can enable hidden indicators by sending the relevant fields directly to the API.

10 Appendix: API Field Reference

Field	Description
`strategy`	Strategy template identifier.
`start/end`	Backtest window in YYYY-MM-DD format.
`capital`	Starting capital in USD.
`fee_bps`	Transaction cost in basis points.
`indicators.policy`	Signal combination rule (any/all/atleast_k).
`indicators.atleast_k`	Minimum signals required when using the at-least policy.
`indicators.max_horizon`	Maximum holding horizon for signals.
`indicators.hist_horizon`	Histogram lookback window.
`indicators.rsi.*`	RSI settings (use, n, rule, oversold, overbought).
`indicators.macd.*`	MACD settings (use, fast, slow, signal, rule).
`indicators.obv.*`	OBV settings (use, rule).
`indicators.ema.*`	EMA cross settings (use, short, long).
`filters.*`	Universe filters (sectors, market-cap bounds, exclusions).