

[Open in app](#)



Search



Write



Member-only story

Fractal Adaptive Moving Average (FRAMA) Strategy? Backtesting using Python and Backtrader



PyQuantLab

Following

6 min read · Jun 15, 2025

21



...

In the intricate world of algorithmic trading, the journey from a promising idea to a deployable, profitable strategy is a multi-stage process of rigorous validation and iterative refinement. The quantitative trading model presented here, centered on the Fractal Adaptive Moving Average (FRAMA), exemplifies the initial research and development phase — a critical step in evaluating a hypothesis rather than deploying a finalized solution. This article delves into the design and preliminary

testing of this FRAMA-based strategy, highlighting its adaptive core and the rationale behind its comprehensive parameterization.



Unlock the Power of Profitable Trading with Code



The Ultimate Algorithmic Trading Bundle

- Over 80 ready-to-run strategies
 - Across 5 powerful categories: momentum, trend-following, mean reversion, ML-based, and volatility
 - Includes step-by-step PDF guides with code explanations, visuals & real-world insights
 - Perfect for traders, quant enthusiasts, and developers
- Everything you need to go from zero to confident algo trader – in one complete package.
- Start building & backtesting proven strategies today:



[Get the Ultimate Bundle](#)

Just Starting Out? Grab the Algo Trading Value Pack

- ⚡ 3 concise eBooks
- 📘 30+ Python-coded strategies
- 🧠 Simple, practical, and actionable — ready to use in Backtrader and real trading platforms

🎯 Ideal if you're looking to learn fast and see results in just a few days.

 [Start with the Value Pack](#)

The Adaptive Heart: Understanding FRAMA

Traditional moving averages suffer from a fundamental trade-off: fast averages are responsive but prone to whipsaws, while slow averages are smoother but lag significantly. The Fractal Adaptive Moving Average (FRAMA), conceived by John Ehlers, seeks to overcome this by dynamically adjusting its smoothing period based on the inherent “fractal dimension” of price data.

At its essence, the fractal dimension quantifies how much a pattern fills space as scale changes. In financial markets, a lower fractal dimension (closer to 1) indicates a trending, “smoother” market, while a higher dimension (closer to 2) signifies a choppy, “rougher” market. FRAMA leverages this insight, adapting its smoothing factor (alpha) to market conditions.

Our implementation of the FRAMA indicator within the Backtrader framework demonstrates this adaptive mechanism:

```
class FRAMAIIndicator(bt.Indicator):
    lines = ('frama', 'fractal_dim', 'alpha',)
    params = (
        ('period', 20),           # Lookback for fractal calculation
        ('slow_period', 200),     # Slow EMA period (choppy)
        ('fast_period', 4),       # Fast EMA period (trending)
    )

    def fractal_to_alpha(self, fractal_dim):
        """Converts fractal dimension to an adaptive smoothing fact
        # Alpha smoothly transitions between fast_alpha (trending)
        slow_alpha = 2.0 / (self.params.slow_period + 1)
        fast_alpha = 2.0 / (self.params.fast_period + 1)
        alpha = fast_alpha + (slow_alpha - fast_alpha) * (fractal_d
        return np.clip(alpha, slow_alpha, fast_alpha)
```

As the code illustrates, a `fractal_dim` closer to 1 (trending) will result in an `alpha` closer to `fast_alpha`, making the FRAMA more responsive. Conversely, a `fractal_dim` closer to 2 (choppy) will yield an `alpha` nearer to `slow_alpha`, resulting in a smoother, less reactive FRAMA. This dynamic behavior is the central hypothesis we aim to validate: can a market-adaptive moving average provide superior signals to static ones?

Constructing the Trading Hypothesis: The FRAMA Strategy

Building upon the adaptive FRAMA, the strategy integrates a multi-layered approach to signal generation, reflecting a comprehensive trading hypothesis. Signals are not solely dependent on simple crossovers but are filtered and confirmed by additional market context.

Key elements of the strategy's logic include:

- Trend Confirmation: Utilizing the slope of the FRAMA to confirm the direction and strength of the underlying trend.
- Price Crossovers: Analyzing the relationship between current price and FRAMA using both “tight” and “wide” thresholds, allowing for different levels of signal confirmation.
- Market Regime Filtering: Employing the fractal dimension itself as a filter, potentially restricting trades to genuinely trending periods, or conversely, identifying mean-reversion opportunities in highly choppy markets.
- Adaptive Thresholds: A more advanced feature where signal thresholds dynamically adjust based on prevailing market volatility, aiming to make the strategy more robust across varying market conditions.
- Fallback Mechanism: A simple moving average crossover strategy serves as a backup, ensuring that trades are still generated even if the FRAMA's complex conditions are not

met. This allows for comparative analysis and ensures that the backtest always yields some trading activity.

The extensive parameterization of the `FRAMAStrategy` is critical for its role as a research tool:

```
class FRAMAStrategy(bt.Strategy):
    params = (
        ('frama_period', 30),           # Lookback for FRAMA fractal c
        ('trend_threshold', 0.01),      # Minimum FRAMA slope for tren
        ('price_threshold_tight', 1.01), # Tight price vs FRAMA cro
        ('fractal_filter', 1.5),        # Max fractal dimension for tr
        ('adaptive_thresholds', True),  # Enable volatility-based th
        ('enable_fallback', True),     # Use SMA crossover if FRAMA s
        ('stop_loss_pct', 0.05),        # Percentage for stop loss
        ('take_profit_pct', 0.5),       # Percentage for take profit
        # ... (numerous other parameters for fine-tuning)
    )

    def generate_frama_signal(self):
        # ... (logic to calculate and apply dynamic thresholds)
        if self.params.adaptive_thresholds and len(self.volatility):
            vol_factor = min(2.0, max(0.5, self.volatility[0] / 0.0
            tight_threshold = 1 + (self.params.price_threshold_tigh
            # ... (apply vol_factor to other thresholds, like trend
```

Each parameter in this configuration represents a testable hypothesis. For example, by varying `fractal_filter`, we can assess the strategy's performance strictly in trending markets versus allowing trades in more ambiguous conditions. The

`adaptive_thresholds` parameter directly investigates the benefit of dynamic sensitivity to market volatility. This flexibility is essential for the iterative tuning process.

The Backtesting Process: Gathering Evidence

To evaluate the initial viability of this strategy, it is put through a simulated trading environment using historical market data. This backtesting phase is crucial for generating preliminary performance metrics and identifying potential strengths and weaknesses.

The Backtrader framework is used to set up the simulation, incorporating realistic trading conditions such as starting capital, commission fees, and position sizing. Standard performance analyzers — Sharpe Ratio, Max Drawdown, and Total Return — are attached to provide a quantitative snapshot of the strategy's hypothetical performance.

```
# Initialize Cerebro (the backtesting engine)
cerebro = bt.Cerebro()
cerebro.addstrategy(FRAMAStrategy)
cerebro.adddata(bt_data) # Add historical data (e.g., BTC-USD)
cerebro.broker.setcash(10000.0) # Starting capital
cerebro.broker.setcommission(commission=0.001) # 0.1% commission
cerebro.addsizer(bt.sizers.PercentSizer, percents=95) # Allocate 95

# Add performance analyzers
cerebro.addanalyzer(bt.analyzers.SharpeRatio, _name='sharpe')
cerebro.addanalyzer(bt.analyzers.DrawDown, _name='drawdown')
```

```

cerebro.addanalyzer(bt.analyzers.Returns, _name='returns')

# Execute the backtest
results = cerebro.run()
strat = results[0]

# Print the results
print(f'Starting Portfolio Value: {cerebro.broker.getvalue():.2f}')
print(f'Final Portfolio Value: {cerebro.broker.getvalue():.2f}')
# ... (Print Sharpe Ratio, Max Drawdown, Total Return with error ha

```



The output from this backtest, including metrics like the Sharpe Ratio and Max Drawdown, offers the first tangible feedback on the strategy's potential. A favorable Sharpe Ratio might suggest reasonable risk-adjusted returns, while a manageable Max

Drawdown indicates resilience. However, these are directional insights only. They guide the next steps of research rather than confirming profitability.

Beyond the First Pass: The Path Forward

It is crucial to reiterate: the current stage represents iteration, not implementation. The FRAMA strategy, as presented, is a sophisticated research tool designed to test a hypothesis about adaptive moving averages. The initial backtest results, regardless of how promising they may appear, are merely data points in a much larger validation process.

The subsequent stages of development typically involve:

- Robustness Testing: Evaluating the strategy's performance across different asset classes, timeframes, and market cycles to ensure its logic holds up universally.
- Parameter Optimization: Systematically searching for optimal parameter sets using techniques like walk-forward analysis, which tests parameters on unseen data to prevent overfitting.
- Sensitivity Analysis: Understanding how stable the strategy's performance is to small changes in parameters.
- Transaction Costs & Slippage: More realistic modeling of real-world trading costs.

- Out-of-Sample Testing: Critically, testing the strategy on data it has never “seen” before to gauge its true predictive power.

In conclusion, this FRAMA-based trading model serves as an excellent framework for quantitative research. Its advanced features, meticulous parameterization, and detailed analytical outputs provide a solid foundation for exploring adaptive trading concepts. However, like all promising ideas in quantitative finance, it remains firmly in the realm of research and development, awaiting further rigorous validation before any consideration for live deployment.

Python

Backtesting

Trading

Finance

Crypto



Written by PyQuantLab

655 followers · 6 following

Following ▾



Your go-to place for Python-based quant tutorials, strategy deep-dives, and reproducible code. For more visit our website: www.pyquantlab.com

No responses yet

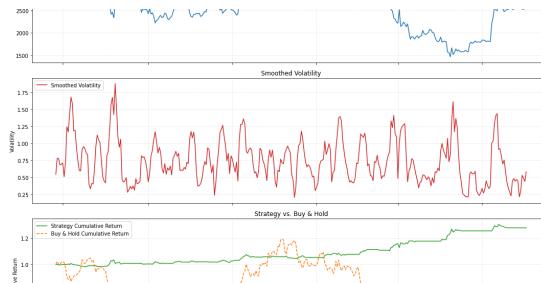


Steven Feng CAI

What are your thoughts?



More from PyQuantLab



PyQuantLab

Volatility Clustering Trading Strategy with Python

Ultimate Algorithmic Strategy Bundle has you covered with over 80 Python...

Jun 3 32 3 ...

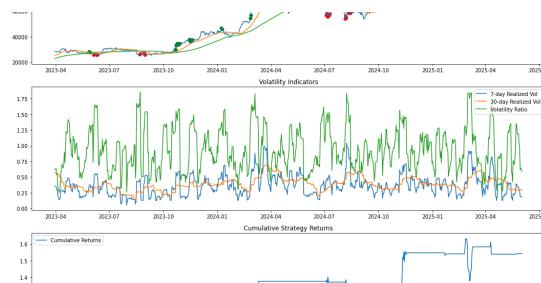


PyQuantLab

An Algorithmic Exploration of Volume Spread Analysis...

Note: You can read all articles for free on our website: pyquantlab.com

Jun 9 54 1 ...



PyQuantLab

Trend-Volatility Confluence Trading Strategy

The Ultimate Algorithmic Strategy Bundle has you covered with over 80...

Jun 3 60 ...



PyQuantLab

Building an Adaptive Trading Strategy with Backtrader: A...

Note: You can read all articles for free on our website: pyquantlab.com

Jun 4 64 ...

See all from PyQuantLab

Recommended from Medium



 MarketMuse

“I Let an AI Bot Trade for Me for 7 Days—It Made \$8,000...”

Subtitle: While you’re analyzing candlestick patterns, AI bots are fron...

 Jun 3  75  3  ...

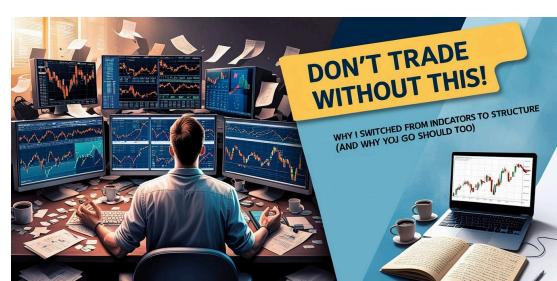
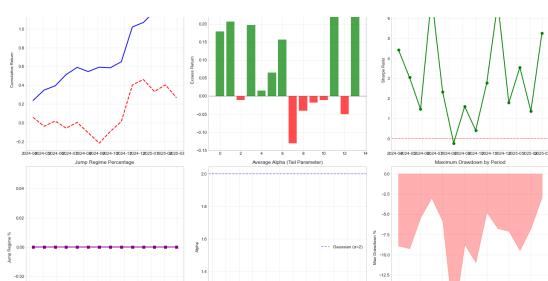


 In DataDrivenInvestor by Mr. Q

Why You Should Ignore Most Backtested Trading...

To be more precise, while the title is limited in length, what I truly mean ar...

 Jun 18  133  2  ...





Capturing Market Momentum with Levy Flights: A Python...

Financial markets are complex systems, often exhibiting behaviors...

Jun 5 102



...



Swapnilphutane

How I Built a Multi-Market Trading Strategy That Pass...

When I first got into trading, I had no plans of building a full-blown system....

6d ago 16 1



...



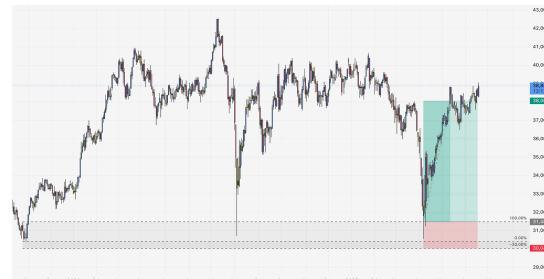
Don't Trade Without This! Why I Switched from...

The Indicator Addiction (That Almost Ruined My Trading)

Jun 12 152 3



...

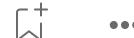


Candence

Exposing Bernd Skorupinski Strategy: How I Profited ove...

I executed a single Nikkei Futures trade that banked \$16,400 with a...

Jun 19 2



...

See more recommendations