NORDIC MACHINE INTELLIGENCE

Nordic Machine Intelligence, vol. 3, pp. 15–110, 2024
Received 8 August 2023 / published 22 May 2024
https://doi.org/10.5617/nmi.10458

Frequency-Histogram Coarse Graining in Elementary and 2-Dimensional Cellular Automata

Sanyam Jain [1], Stefano Nichele [2,3]

1. *Dept. of Computer Science and Communication, Østfold University College, Halden, Norway*
2. *Dept. of Computer Science, Oslo Metropolitan University, Oslo, Norway*
3. *Dept. of Computer Science and Communication, Østfold University College, Halden, Norway*

Abstract
Cellular automata and other discrete dynamical systems have long been studied as models of emergent complexity. Recently, neural cellular automata have been proposed as models to investigate the emerge of a more general artificial intelligence, thanks to their propensity to support properties such as self-organization, emergence, and open-endedness. However, understanding emergent complexity in large scale systems is an open challenge. How can the important computations leading to emergent complex structures and behaviors be identified? In this work, we systematically investigate a form of dimensionality reduction for 1-dimensional and 2-dimensional cellular automata based on coarse-graining of macrostates into smaller blocks. We discuss selected examples and provide the entire exploration of coarse graining with different filtering levels in the appendix (available also digitally at this link: https://s4nyam.github.io/eca88/). We argue that being able to capture emergent complexity in AI systems may pave the way to open-ended evolution, a plausible path to reach artificial general intelligence.

Keywords: Artificial Intelligence; Cellular Automata; Complexity; Artificial Life; Coarse Graining

Introduction

Recent advances in deep learning, such as large language models, have shown promise in some tasks thought to be possible only in biological intelligence. However, such systems remain rather rigid if compared to the adaptivity and the open-endedness of living organisms [1]. In fact, natural intelligence may be considered a form of spontaneously self-organizing and ever evolving complex system producing increasing levels of emergent complexity [2]. Clune [3] describes the AI generating algorithm approach as a possible alternative to creating machines that can "automatically learn how to produce general AI", as opposed to the "manual AI approach" which is predominant in the AI community nowadays. *Natural evolution is the only generating algorithm that has been able to produce general intelligence.* Artificial Life frameworks such as Cellular Automata (CA) have long been used as models to understand computation, life, and evolution.

Recent advances of such models, including continuous CA [4, 5] and neural CA [6, 7], have been proposed as substrates to study the emergence of a general intelligence [8]. However, an open question is how to quantify the emergent complexity of such artificial systems at different scales, in order to guide evolution towards higher emergent complexity. In this work, we consider Elementary Cellular Automata (ECA), 2D CA based on Game of Life rules, and an evolved multiple neighborhoods CA (MNCA) rule. We perform a dimensionality reduction process using a coarse-graining technique [9], in order to identify complex pattern formations and highlight them while filtering out uninteresting computations happening in the background. We utilize a frequency-based technique that scans the space-time diagrams and collects frequencies of different blocks, which are then used for coarse-graining by effectively replacing blocks with frequency below a certain threshold with a meaningful state, in fact reducing the dimensionality of the CA. Additionally, such coarse-graining technique may allow identifying emergent complexity at different scales. We highlight some example results in our discussion (while leaving the full results for the Appendix), with the aim of discussing the implications of this approach for the long-term goal of understanding how AI systems may become more open-ended, adaptive, and general, through the emergence of complex behaviors out of simple rules.

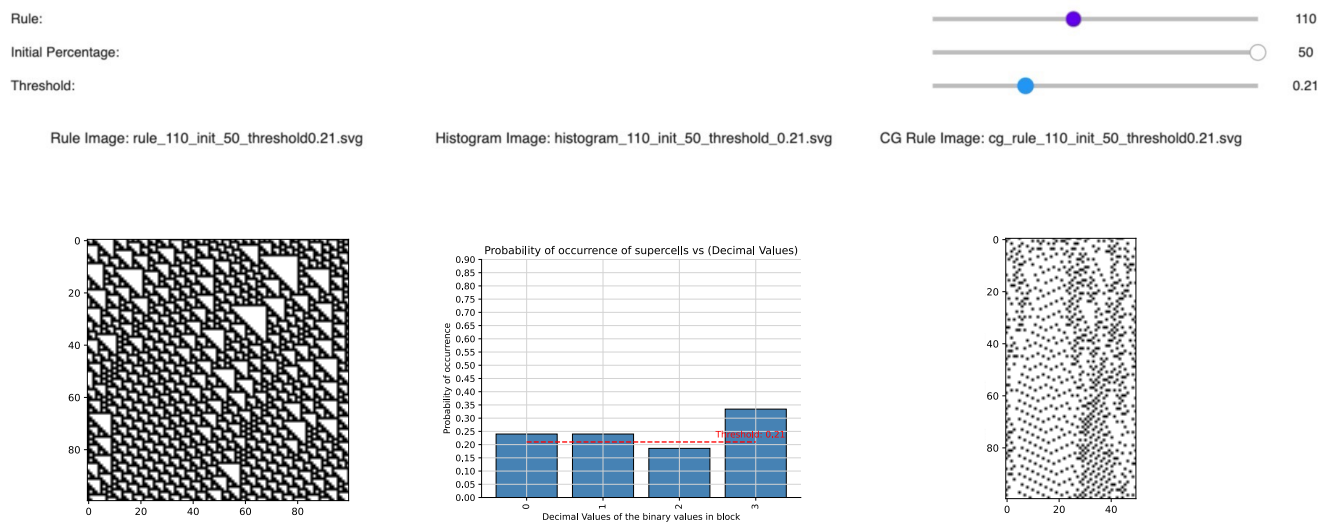

Figure 1: Snapshot from the platform to study Binary FHCG for ECA. In this example, Rule 110 is depicted on the left. In the centre, the frequency histogram for blocks of size 2x1 is shown. On the right, the resulting coarse graining with the selected threshold. The rule, the initial state, and the threshold can be selected from the sliders at the top.

Background and Related Work

Cellular Automata

Cellular Automata are computational models characterized by a regular grid of cells, each capable of existing in a discrete number of states. CA evole through discrete time steps guided by simple rules dependent on neighboring cell states. Among them, Elementary Cellular Automata (ECA) stand out with their one-dimensional grid, featuring binary cell states (on or off). Despite their simplicity, ECAs exhibit remarkable complexity [10] as the future state of each cell relies solely on its current state and the states of its two neighboring cells, leading to the emergence of intricate patterns. Stephen Wolfram [11] classified their behavior in four qualitative classes: 1) evolving to homogeneous states; 2) evolving to simple and periodic patterns; 3) evolving to disordered patterns; 4) evolving to complex localized structures. It has been demonstrated that, due to different types of equivalences, there exist a set of 88 unique ECA rules (out of the 256 possible rules) [12].

The Game of Life (GoL) [13] is one of the most renowned CA instances, operating on a two-dimensional grid with cells in either alive or dead states. GoL evolves through discrete generations based on specific rules involving the eight adjacent neighbors' states, leading to intriguing phenomena such as self-replicating structures and gliders. Advancing further, Multi-Neighbor Cellular Automata (MNCA) [14] enable interactions with a broader range of neighboring cells, allowing simulations of complex phenomena like urban growth patterns, traffic flow, and ecological dynamics, where local interactions with multiple neighbors significantly influence macroscopic behavior and emergent properties. These diverse branches of cellular automata find applications across disciplines, offering valuable insights into complexity and emergent behavior, showcasing how elegance and simplicity can yield a rich tapestry of patterns and behaviors and contributing to our understanding of complex systems in both natural and artificial domains [10].

Complexity and Open-Endedness

Gaining insight into the emergent computation of a complex system is not trivial, in particular by looking at the activity of the simple components of the system. In [15], Melanie Mitchell explains how a "computational mechanics" framework may be employed to explain the emergent complexity of a CA. In CA, particles and their interactions may describe the computation embedded in the space-time behavior, providing insights into how the decentralized processing happens. Mitchell argues that "the language of particles and their interactions form an explanatory vocabulary for decentralised computation in the context of one-dimensional cellular automata". One popular way to approximate the behavior of a CA and assess its complexity is through compression. For example, in [16] they partition the 1D CA space through compression. In [17], compression is used also for 2D and 3D CA. However, compression as proxy for complexity has some issues, such as the fact that random or chaotic states are the most incompressible, while complex localized patterns have somewhat "intermediate" levels of compressibility. For a review of different proposed complexity metrics for CA and other complex systems, please refer to [18], where the notion of complexity as degree of hierarchy is of particular interest when complexity emerging at different scales is of relevance. One notable example of equal complexity at different hierarchical scales is the implementation of the Game of Life inside the Game of Life [19]. Recent works [9, 20, 21] show more advanced methods to capture and evaluate

complex emerging behaviour. Such works propose multiple approaches for both discrete CA and continuous CA. In particular, the authors investigate coarse-graining and artificial neural networks (Auto-Encoders) to quantify complexity, conducting experiments to demonstrate the metric's ability to automatically identify CA with complex emerging patterns. The coase-graining method proposed in [9], based on frequency-histograms, has been shown to qualitatively approximate selected ECA rules to the particle domain boundaries presented in [22, 18].

In this work, we systematically explore frequency-histogram coarse graining for the entire ECA space, 2D Game of Life, and an example of MNCA.

Algorithm 1 Auto Thresholds & processing FHCG of 1D Cellular Automata

Data: State is the evolved ECA at current time step, p_i is probability of a cell i, α is threshold, $\text{count}_G(s_i)$ is count of occurrence of supercell i in state G

Result: Modified Automatons

foreach *state* **do**

  calculate chunks on current state using block size

  store chunks as supercells

end

▷ At this stage you should have collection of all chunks for ECA state

foreach *supercell* **do**

  $\text{p}_i = \frac{\text{count}_G(s_i)}{\sum_{j \in S(\text{in})} \text{count}_G(s_j)}$

end

▷ Store unique Probabilities of each block type, block types are shown below

foreach α *in Thresholds* **do**

  $\text{G(i)} = \begin{cases} 0, & \text{if } p_i \geq \alpha \\ 1, & \text{if } p_i < \alpha \end{cases}$

end

Frequency-Histogram Coarse Graining

In order to capture and possibly quantify emergent complexity in dynamical systems, it is beneficial to reduce large systems into smaller sizes, while being able to preserve important pattern formations and filter out background computations.

The work in [9] introduces different coarse graining methods, namely Frequency-Histogram Coarse Graining (FHCG), Clustering, and Auto-Encoder based Coarse Graining. Of particular interest due to its computational simplicity is FHCG, where blocks of cells are mapped to supercell states based on the probability of their occurrence in the CA space-time evolution. The frequency of different configurations is computed and used to create a frequency histogram. Supercells are then assigned new states based on their frequencies, with a threshold determining the mapping to either state 0 or 1 (in case of binary coarse graining). It also allows for multiple-state coarse graining.

Proposed Methods

We propose a new way to implement ECA with FHCG such that emerging artefacts of ECA can be closely analysed using FHCG. While we focus on most of the results for analysis in ECA, the same methodology also works for 2D CA (or more advanced Neural CA). Initially, the evolved CA rule is chunked into small pieces of fixed size. These chunks are akin to supercells and hence can be assumed as a single entity which is further replaced with a single cell. The way we replace these supercells with one single cell is exactly what is calculated by FHCG algorithm. Once the supercells are available for the complete CA, the probability of occurrence for each of those is calculated and replaced in the CA. For example, in ECA a block of fixed size two would have four different kind of binary occurrences (00, 01, 10, 11). For each of those binary macro-symbols, the occurrence probabilities are calculated. In other words, for all the four different supercells, the corresponding occurrence count is calculated (and hence the probability of each supercell). As supercells are replaced based on their probabilities, we then use a filtering threshold such that values higher than the threshold get eliminated (replaced by 0) and below are kept, i.e., the corresponding supercells stays active (replaced by 1). Simply put, in a CA with large occurrence of a certain type of cell pairs will yield higher probability of occurrence of their corresponding supercells. By varying the thresholds, the density of the particles (and hence the attractor) can be adjusted, while varying the block size will control the magnification.

Algorithm 1 explains the simple working of FHCG in ECA. The Algorithm takes the current state of the ECA as input, then scans each step of the evolved ECA to create chunks of fixed block size. These binary blocks are sorted to get correct labels and decimal value for the histogram generation. Further, probabilities of occurrence of the supercells are calculated and stored. These probabilities are replaced with the available states (here *0* and *1*) accordingly to the chosen threshold. Finally, different coarse grained states of the original CA is computed.

We use this proposed FHCG technique to study complex behaviour in ECA for the unique 88 rules (by considering their Minimum Equivalence [23]). We propose a novel dataset consisting of all 88 rules visualised using automatic thresholding for FHCG and corresponding CA output. Automatic thresholding decides how to set the thresholds on the occurrence of probabilities for the frequency histogram. For each of the four possible blocks (for a block size 2 applied on ECA), we add a fraction δ to the highest probability value and assign it as one of the three thresholds. A snapshot for the proposed method to study ECA is shown in Figure 1.

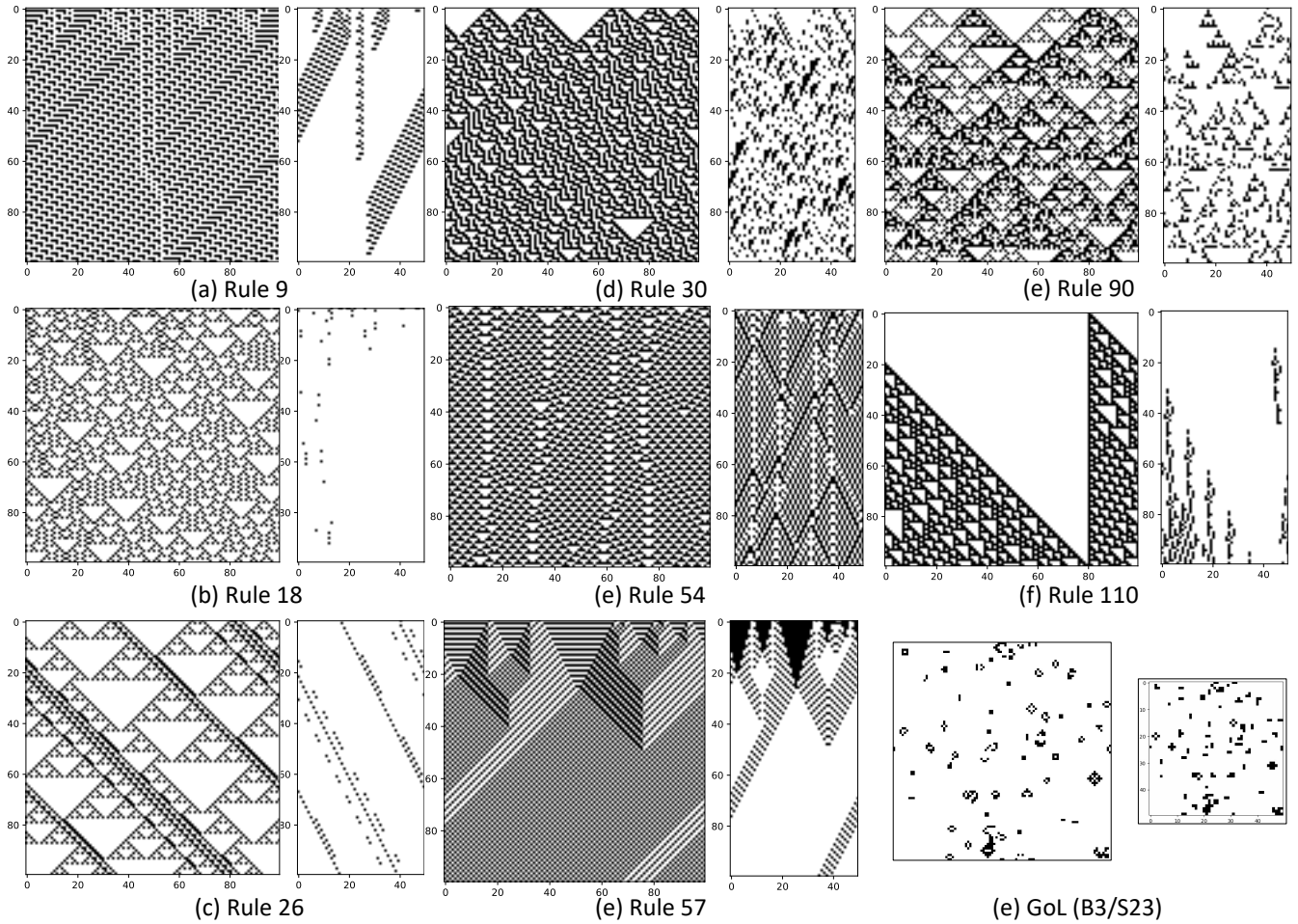

Figure 2: Example results from selected rules. Left: space-time diagram; Right: coarse-grained space-time diagram.

Discussion of Results and Conclusion

Our proposed coarse graining method is available at https://s4nyam.github.io/mncaportal. An extract of our full results (presented in the Appendix) is depicted in Figure 2. Our method of FHCG effectively compresses the cellular automata representations while highlighting rare events, making it easier to study interesting emergent behaviors and their complexity. Such understanding is crucial for studying emergent phenomena and understanding the evolution in cellular automata and other dynamical systems.

In particular, it can be seen that hidden structures can be discovered in the underlying computation for several rules, such as rule 18, 30, and 90 (class III) or rule 110 (class IV). In rule 54 (class IV), it can be easily seen that diagonal particles are discovered. For rules with simple periodic patterns as rule 9, 26, or 57 (class II), the background computations are filtered our and the emergent patterns are extracted. In case of two-dimensions, the gliders and other emergent computational structures in the Game of Life, are simplified in a lower dimension. While some granularity is lost, it can be easily identified where the computations are ongoing (for example a glider may be replaced by a more compact structure). While for 1D CA, the entire space-time diagram can be visualized in 2D, for 2D CA a sequence of 2D snapshots would be needed (in Figure 2 we provide a single snapshot), or alternatively a video sequence.

In general, coarse graining allows extracting valuable insights from CA computations. In particular, it provides an approximation that enables to speed-up calculations related to attractors or pattern matching. Additionally, it makes possible to highlight relevant irregularities or in other words rare but important computations. Finally, reducing the dimensionality of large scale systems may allow observing emergent patterns of behavior not readily accessible at the level of individual agents or cells. Coarse graining methods extract the essential components of the available information, by distilling the space-time diagram.

These techniques can be applied recursively and in combination to study such systems at various scales, providing valuable insights into their emergent properties and overall complexity. Being able to identify and quantify emergent complexity for large-scale evolving systems may be a crucial step in achieving open-endedness. Novel ways of guiding evolution and learning towards higher complexity levels may be discovered, ultimately allowing our artificial systems to become more adaptive.

References

1. Stanley KO. Why open-endedness matters. Artificial life 2019; 25:232–5
2. Booker L. Perspectives on adaptation in natural and artificial systems. Vol. 8. Santa Fe Institute Studies on, 2005
3. Clune J. AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. arXiv preprint arXiv:1905.10985 2019
4. Chan BWC. Lenia-biology of artificial life. arXiv preprint arXiv:1812.05433 2018
5. Chan BWC. Lenia and Expanded Universe. *The 2020 Conference on Artificial Life*. MIT Press, 2020
6. Mordvintsev A, Randazzo E, Niklasson E, and Levin M. Growing neural cellular automata. Distill 2020; 5:e23
7. Variengien A, Nichele S, Glover T, and Pontes-Filho S. Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. arXiv preprint arXiv:2106.15240 2021
8. Gregor K and Besse F. Self-Organizing Intelligent Matter: A blueprint for an AI generating algorithm. 2021. arXiv: 2101.07627 [cs.NE]
9. Cisneros H, Sivic J, and Mikolov T. Visualizing computation in large-scale cellular automata. *The 2020 Conference on Artificial Life*. MIT Press, 2020
10. Wolfram S et al. A new kind of science. Vol. 5. Wolfram media Champaign, IL, 2002
11. Wolfram S. Universality and complexity in cellular automata. Physica D: Nonlinear Phenomena 1984; 10:1–35
12. Wolfram S. Theory and applications of cellular automata. World Scientific 1986
13. Berlekamp ER, Conway JH, and Guy RK. Winning ways for your mathematical plays, volume 4. AK Peters/CRC Press, 2004
14. Kraakman B. Understanding Multiple Neighborhood Cellular Automata. 2021
15. Mitchell M, Crutchfield JP, Das R, et al. Evolving cellular automata with genetic algorithms: A review of recent work. *Proceedings EvCA'96*. Vol. 8. Moscow. 1996
16. Zenil H. Compression-based investigation of the dynamical properties of cellular automata and other systems. Complex Systems 2010; 19:1–28
17. Nichele S and Tufte G. Measuring Phenotypic Structural Complexity of Artificial Cellular Organisms: Approximation of Kolmogorov Complexity with Lempel-Ziv Compression. *Innovations in Bio-inspired Computing and Applications: Proceedings of the 4th International Conference on Innovations in Bio-Inspired Computing and Applications, IBICA 2013, August 22-24, 2013-Ostrava, Czech Republic*. Springer. 2014 :23–35
18. Mitchell M. Complexity: A guided tour. Oxford university press, 2009
19. Bradury P. Life in Life. Youtube. 2012. Available from: https://youtu.be/xP5-ileKXE8
20. Cisneros H, Sivic J, and Mikolov T. Evolving structures in complex systems. *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2019 :230–7
21. Jain S, Shrestha A, and Nichele S. Capturing emerging complexity in lenia. *Italian Workshop on Artificial Life and Evolutionary Computation*. Springer. 2023 :41–53
22. Hanson JE and Crutchfield JP. Computational mechanics of cellular automata: An example. Physica D: Nonlinear Phenomena 1997; 103:169–89
23. Castillo-Ramirez A and Magaña-Chavez MG. A study on the composition of elementary cellular automata. arXiv preprint arXiv:2305.02947 2023
24. Medernach D, Kowaliw T, Ryan C, and Doursat R. Long-term evolutionary dynamics in heterogeneous cellular automata. *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. 2013 :231–8

Appendix

In the following pages of the Appendix, we are presenting a systematic investigation of coarse-graining for:

- The 88 Minimum Equivalent ECA rules
- The Game of Life
- An evolved MNCA rule.

Results for ECA are also available digitally at the following link:

- https://s4nyam.github.io/eca88.

A portal which provides the pointers to all our code implementations is available at the following link:

- https://s4nyam.github.io/mncaportal.

Starting from the next page, we present one rule per page. Each page is structured as follows:

- The first row and first image displays the original CA simulation with 1% initialisation. The CA size is 100, and the rule is applied for 100 steps.
- Further, the second image shows the histogram plot for each of the blocks, containing three different thresholds. Note that the used block size is 2x1 for 1D CA and 2x2 for 2D CA.
- Further, the next three images show the coarse-grained versions of the original CA simulation with the corresponding thresholds applied in ascending order.
- Similarly, for the second and third rows, the density of initialisation is 20% and 50% respectively.

Note that for some rules, not all blocks are present in the space-time diagram and therefore there are less bars in the histogram and less coarse-graining results visualized.

The thresholds are obtained by adding a δ of 0.005 to the calculated probabilities. Please note that for some rules, the resulting threshold includes two histogram bars with almost identical values. This results in less coarse-graining to visualize. An example is Rule 43 (Third row with 50% initialisation in Table 38), where several bars have almost identical frequency.

The following Appendix in PDF is also available at the following link:

- https://s4nyam.github.io/mncaportal/appendix.pdf

Table 1: FHCG plots for ECA Rule 0.

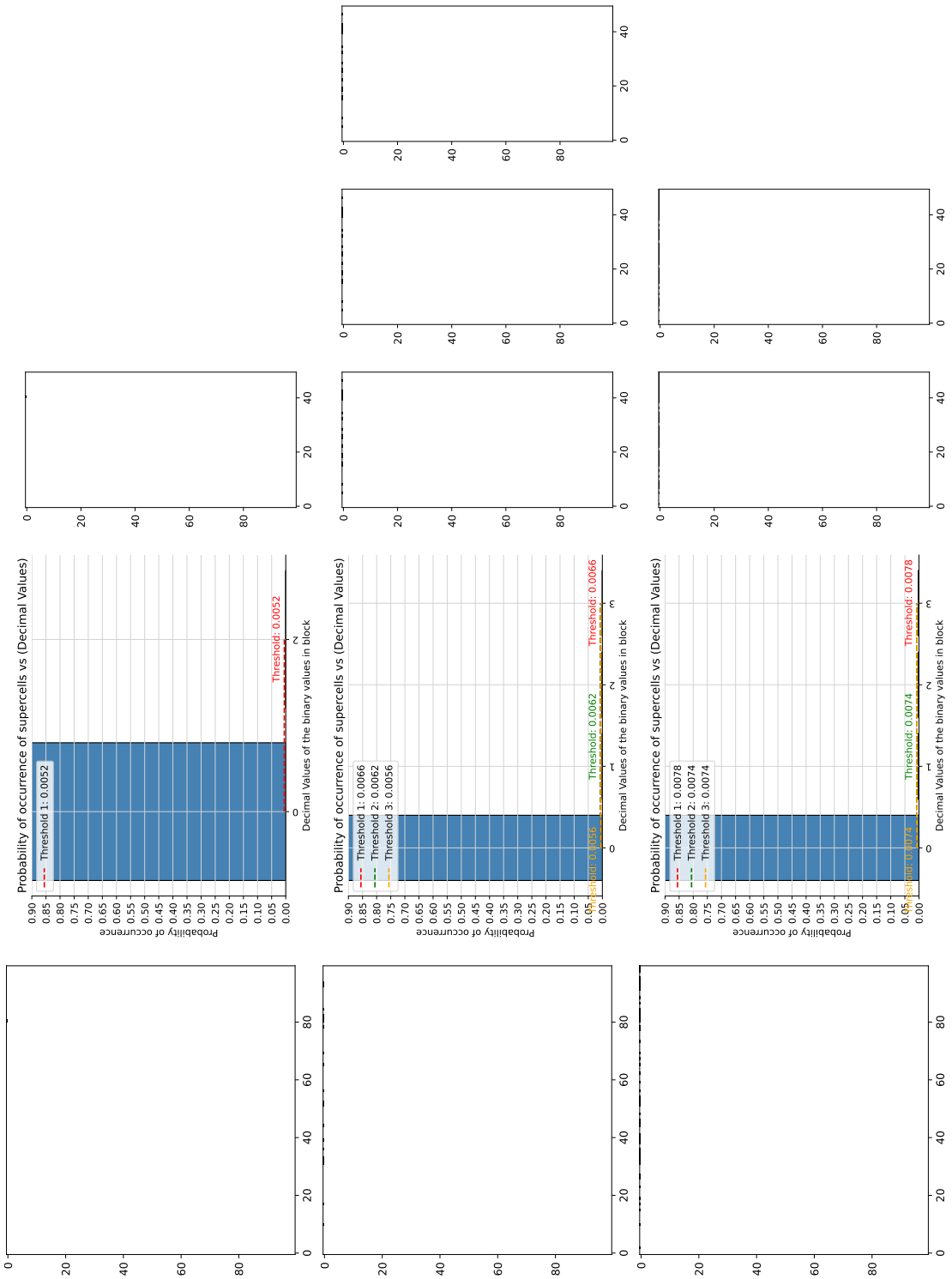

Table 2: FHCG plots for ECA Rule 1.

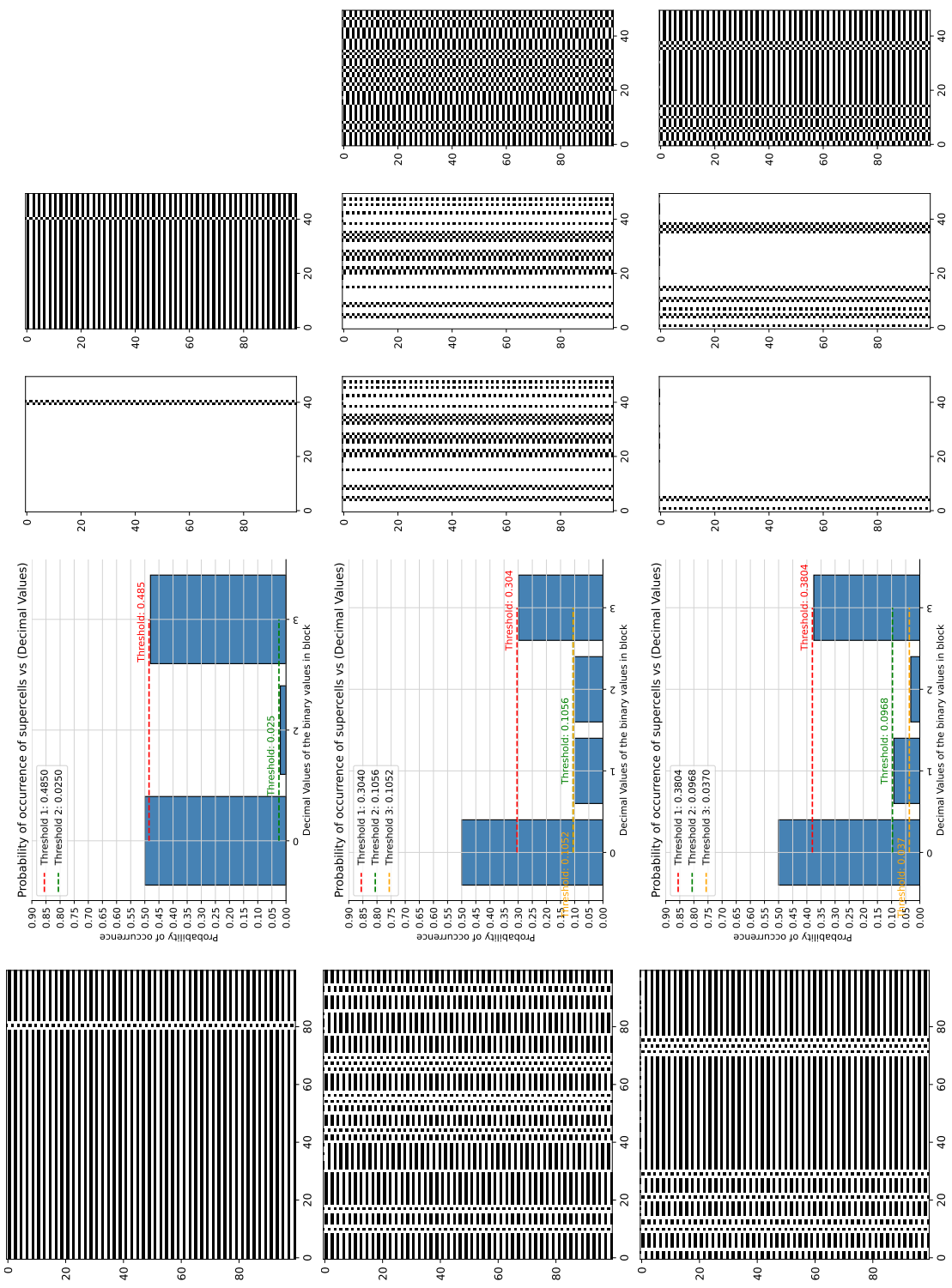

Table 3: FHCG plots for ECA Rule 2.

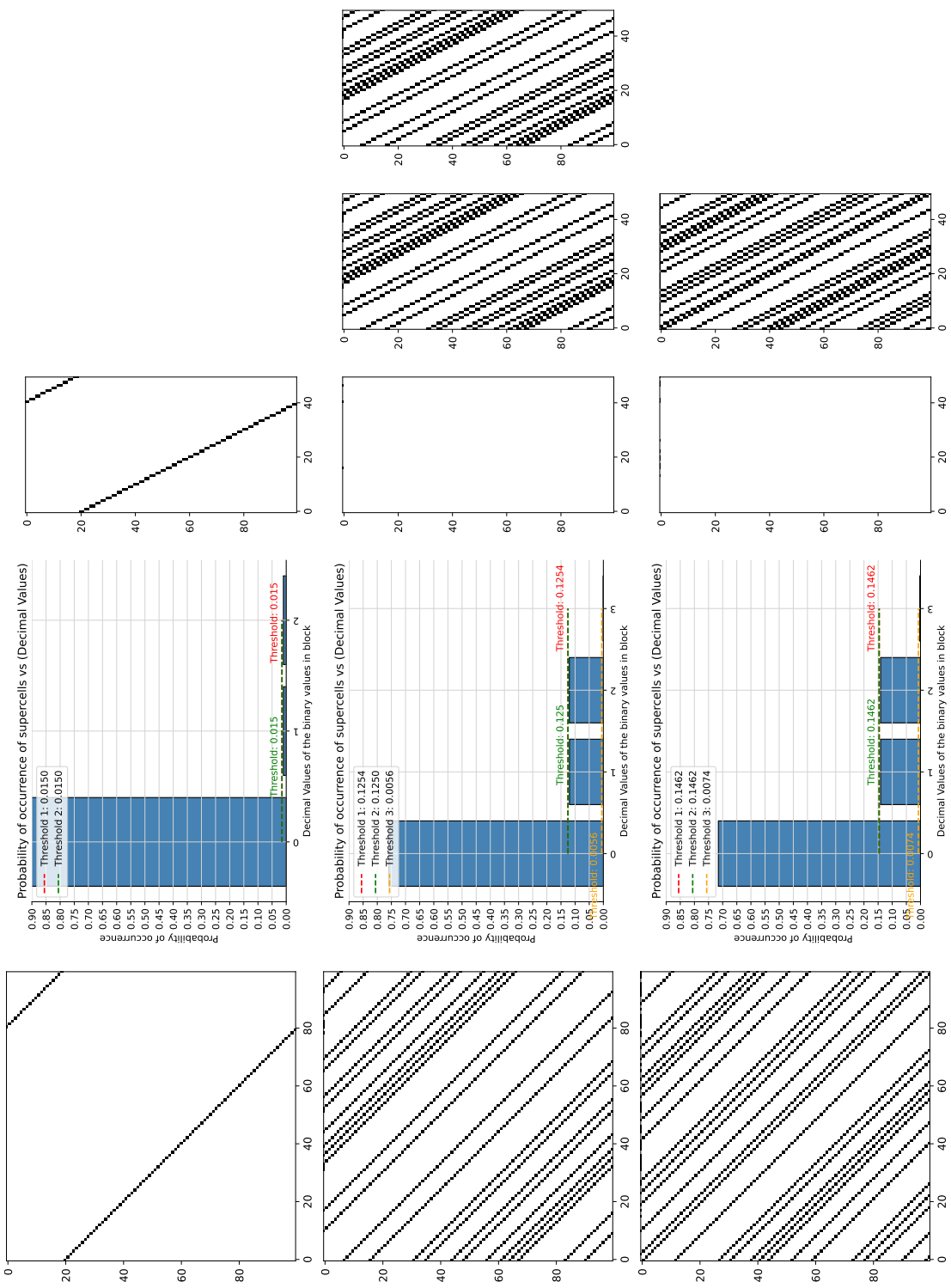

Table 4: FHCG plots for ECA Rule 3.

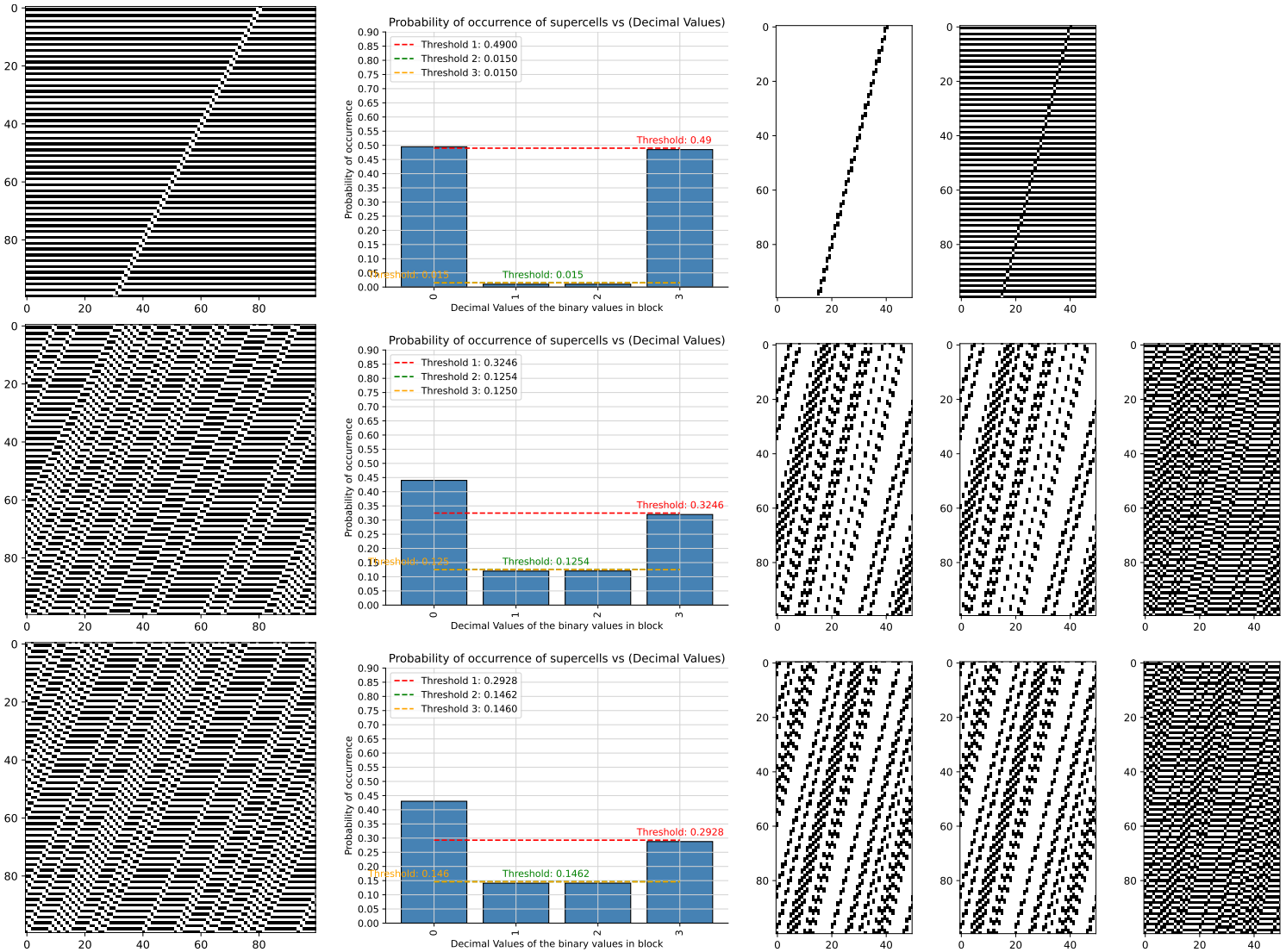

Table 5: FHCG plots for ECA Rule 4.

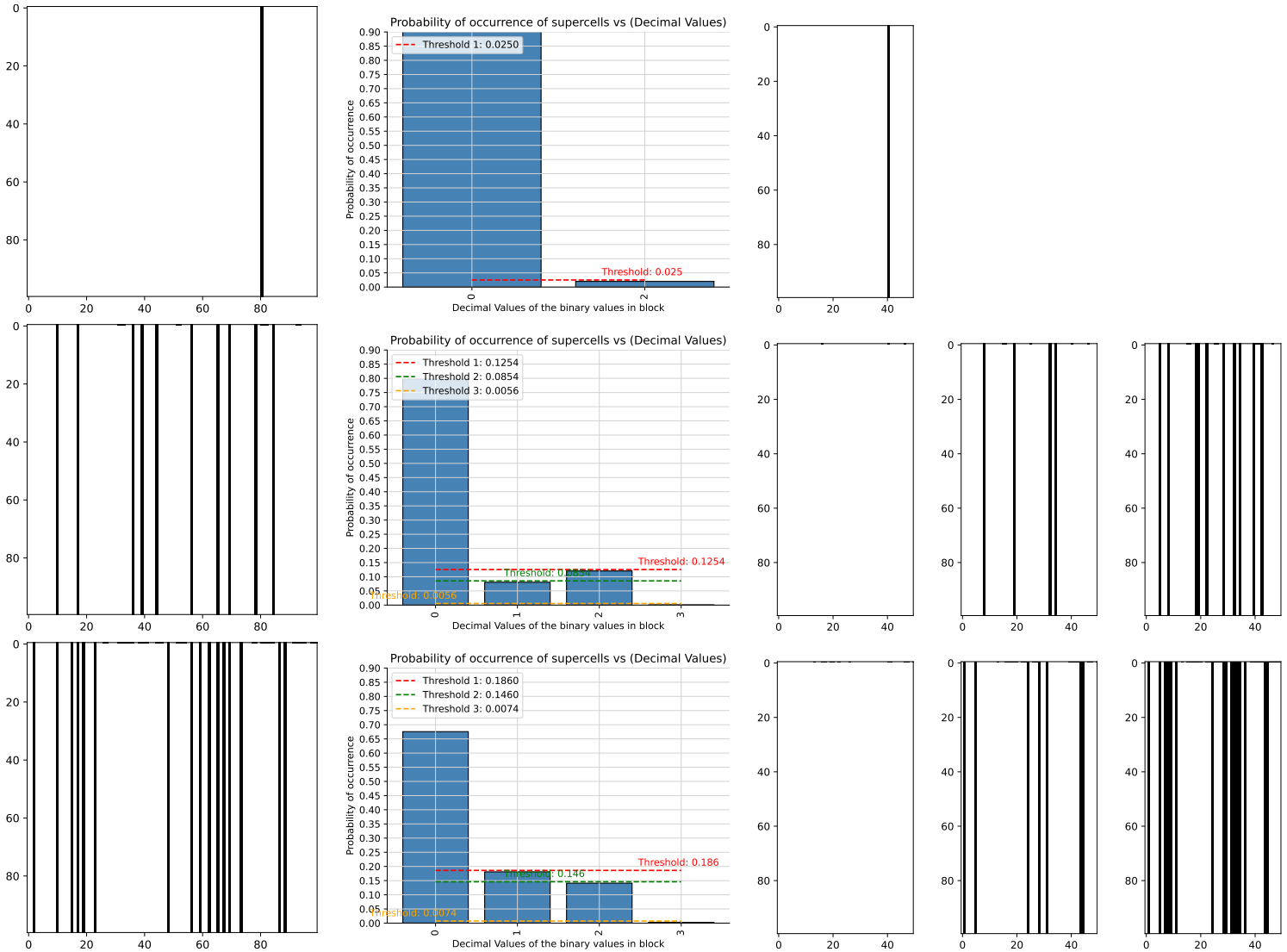

Table 6: FHCG plots for ECA Rule 5.

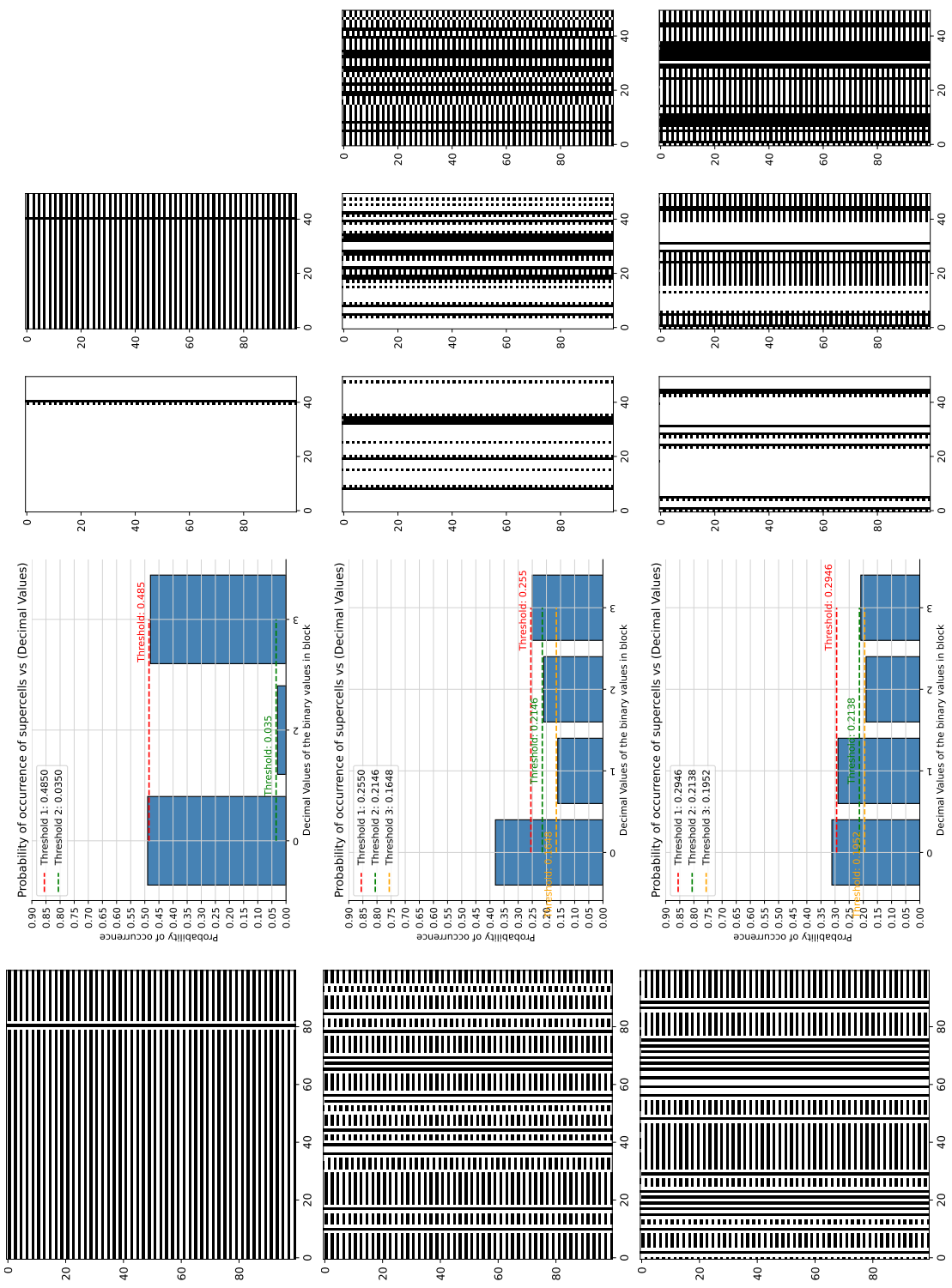

Table 7: FHCG plots for ECA Rule 6.

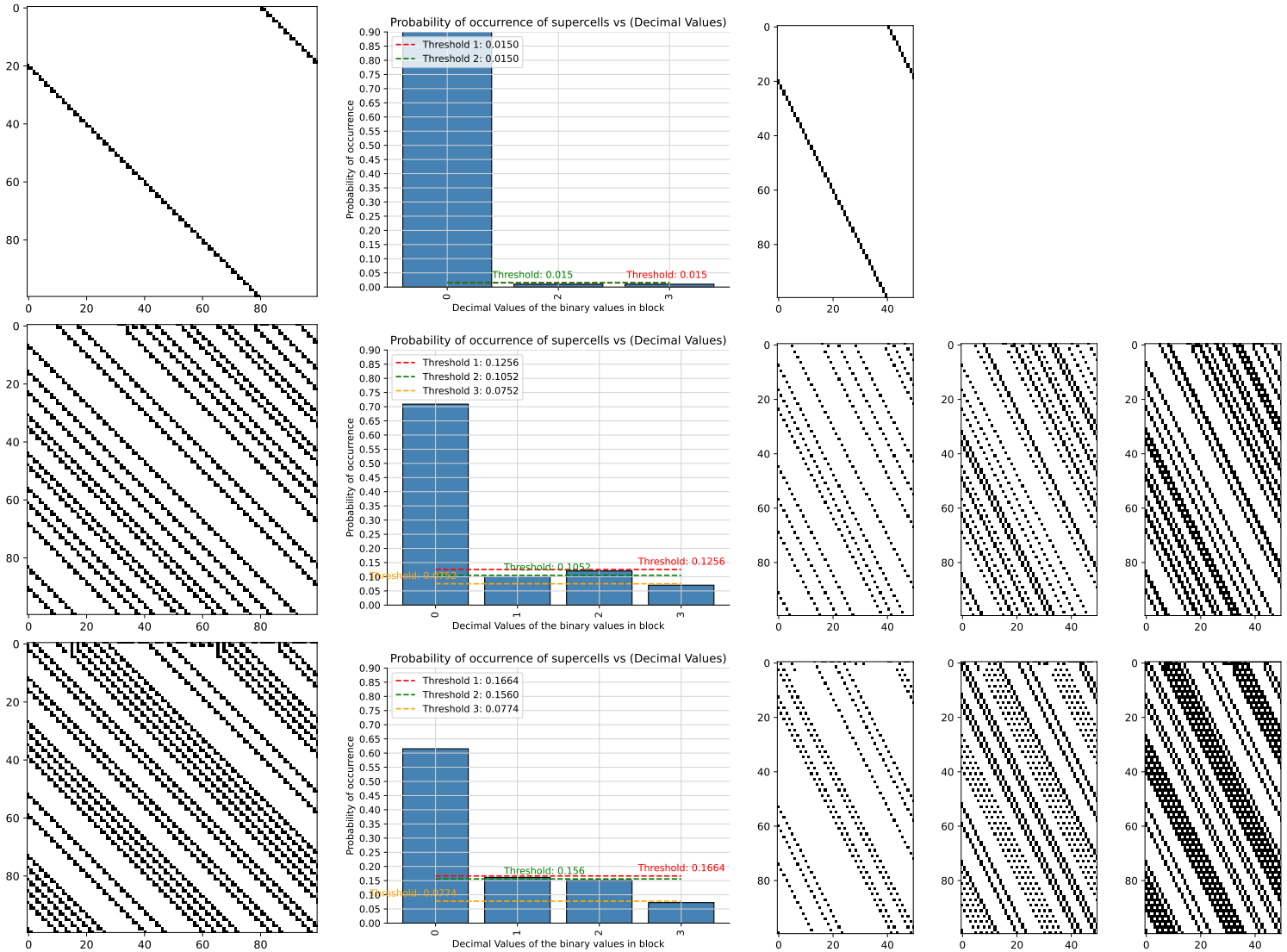

Table 8: FHCG plots for ECA Rule 7.

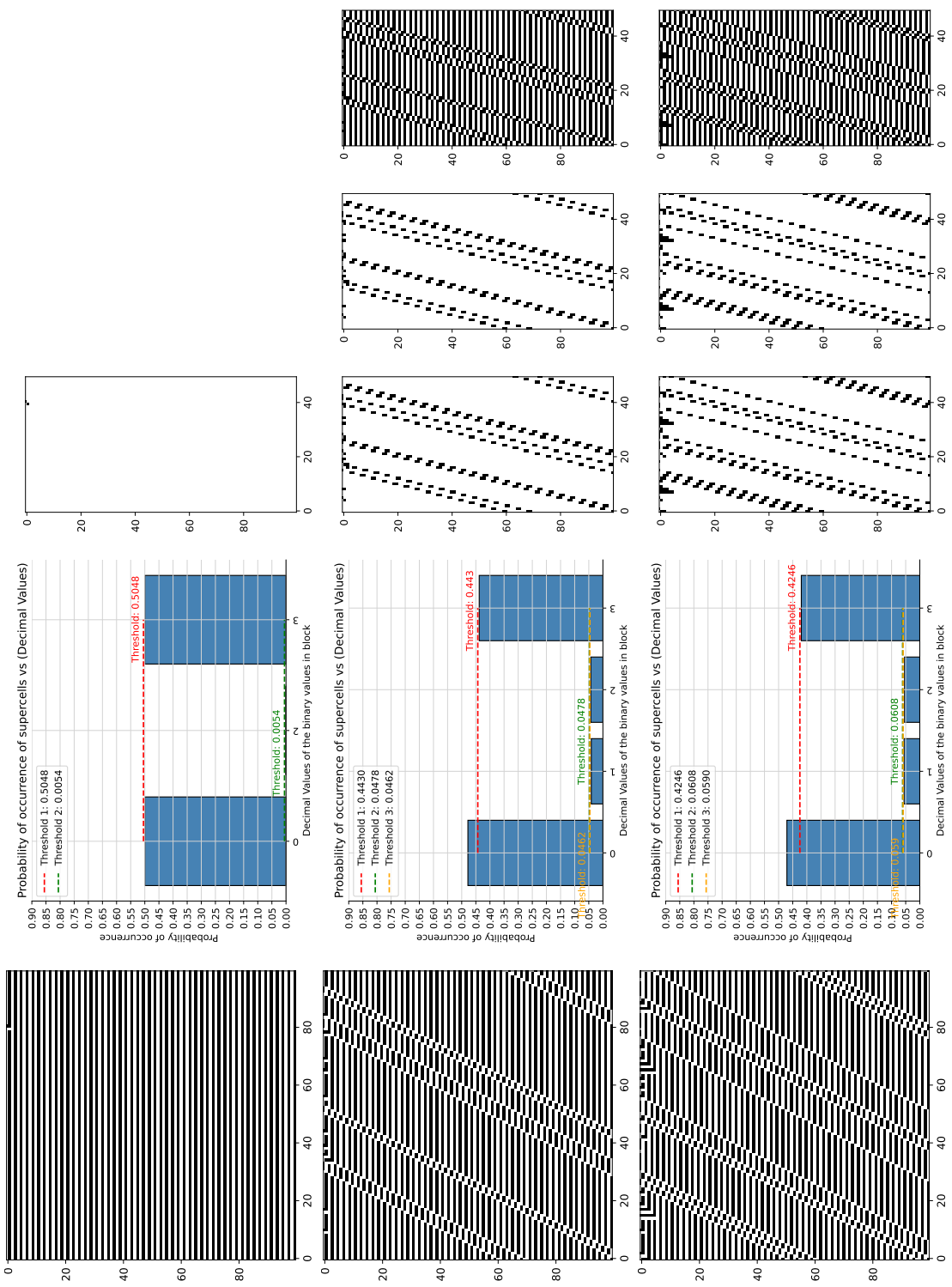

Table 9: FHCG plots for ECA Rule 8.

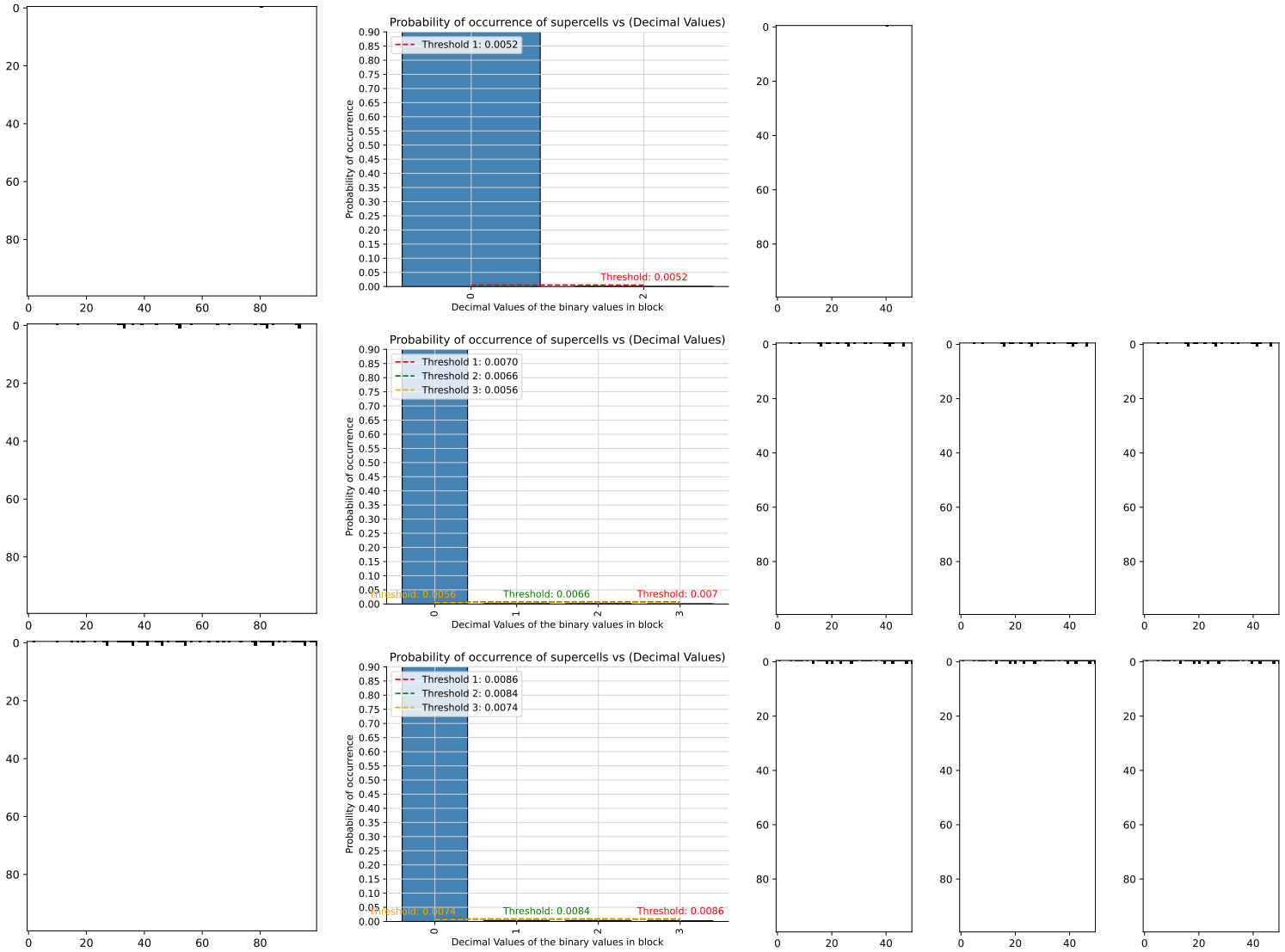

Table 10: FHCG plots for ECA Rule 9.

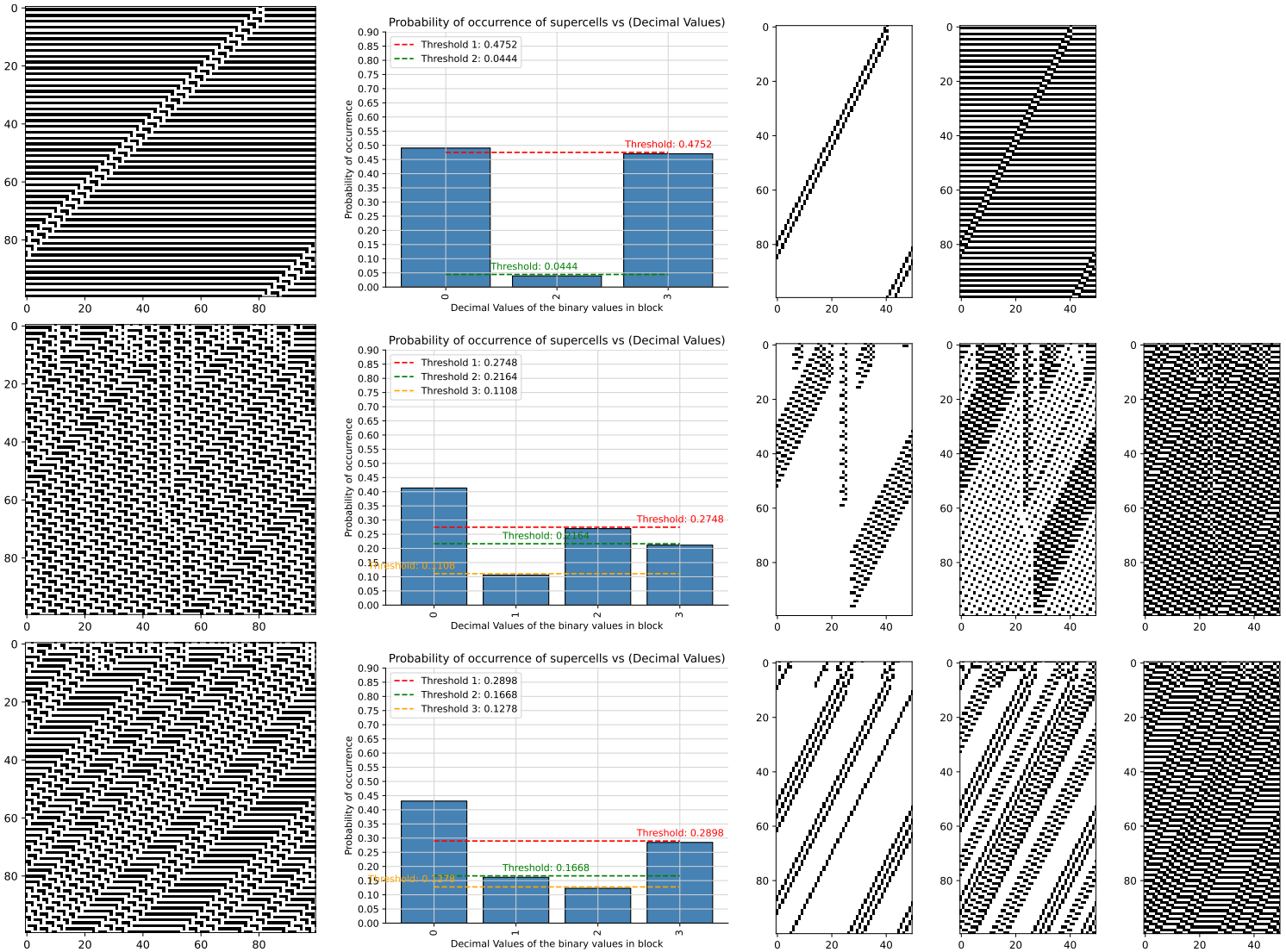

Table 11: FHCG plots for ECA Rule 10.

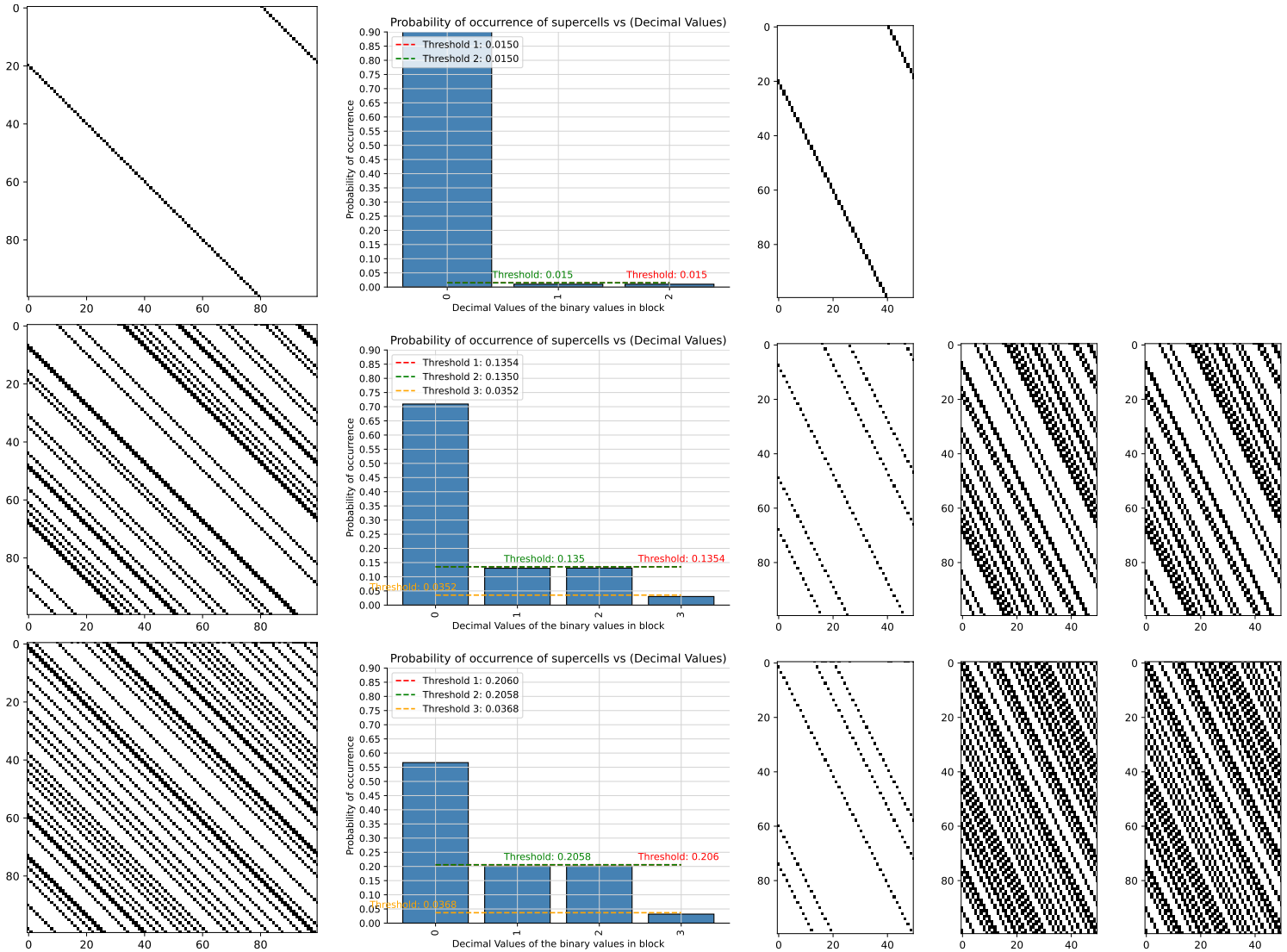

Table 12: FHCG plots for ECA Rule 11.

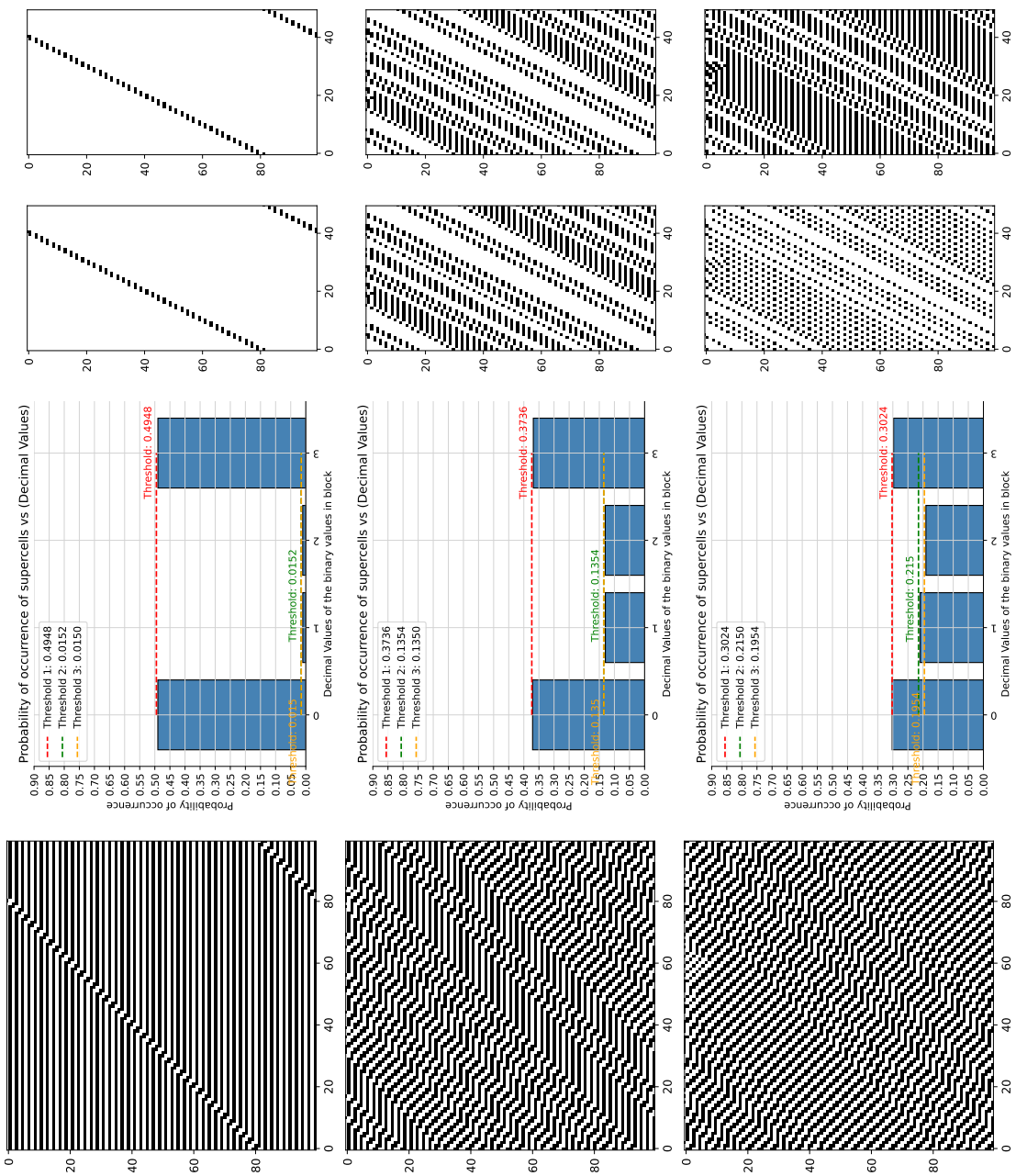

Table 13: FHCG plots for ECA Rule 12.

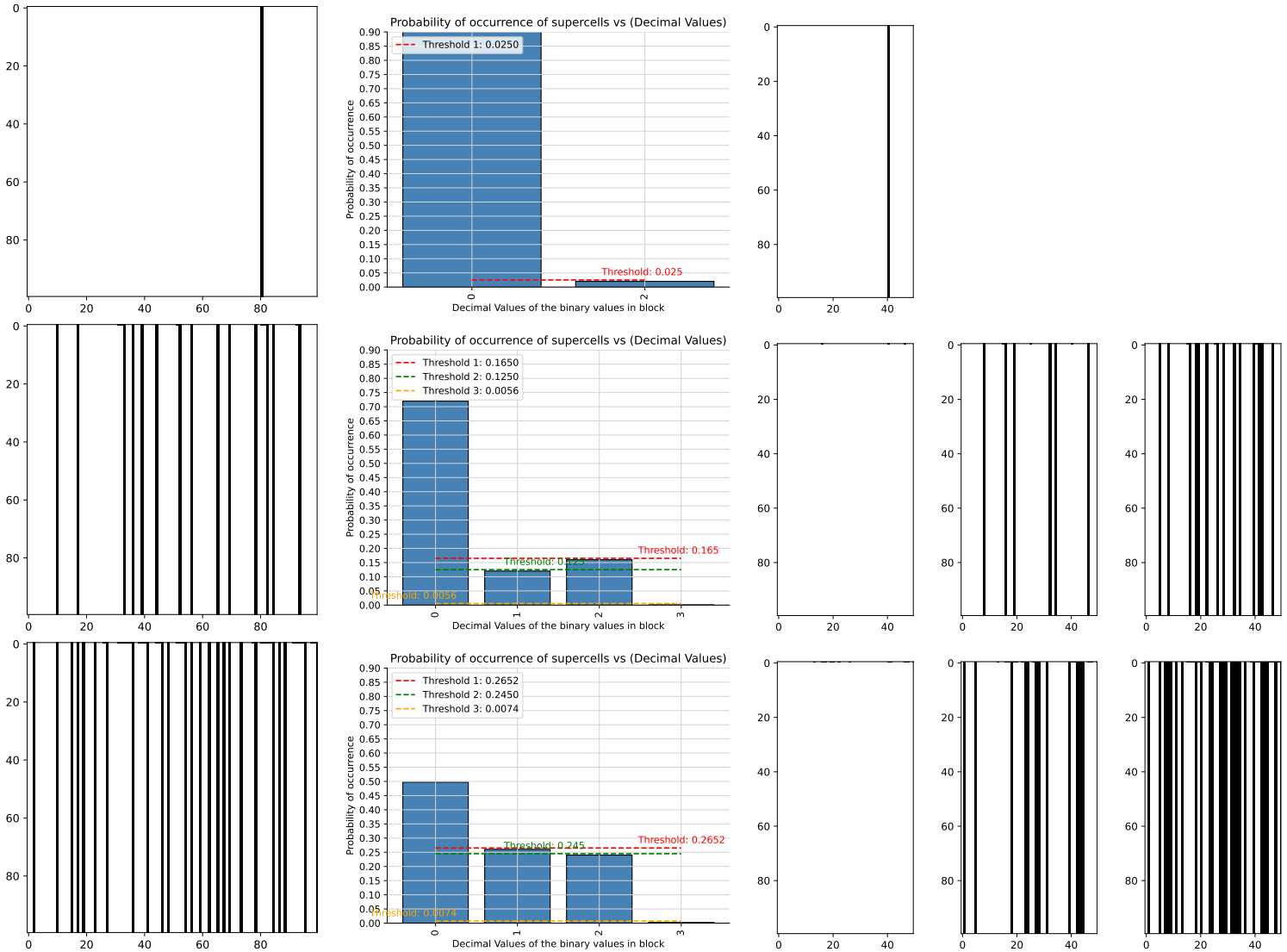

Table 14: FHCG plots for ECA Rule 13.

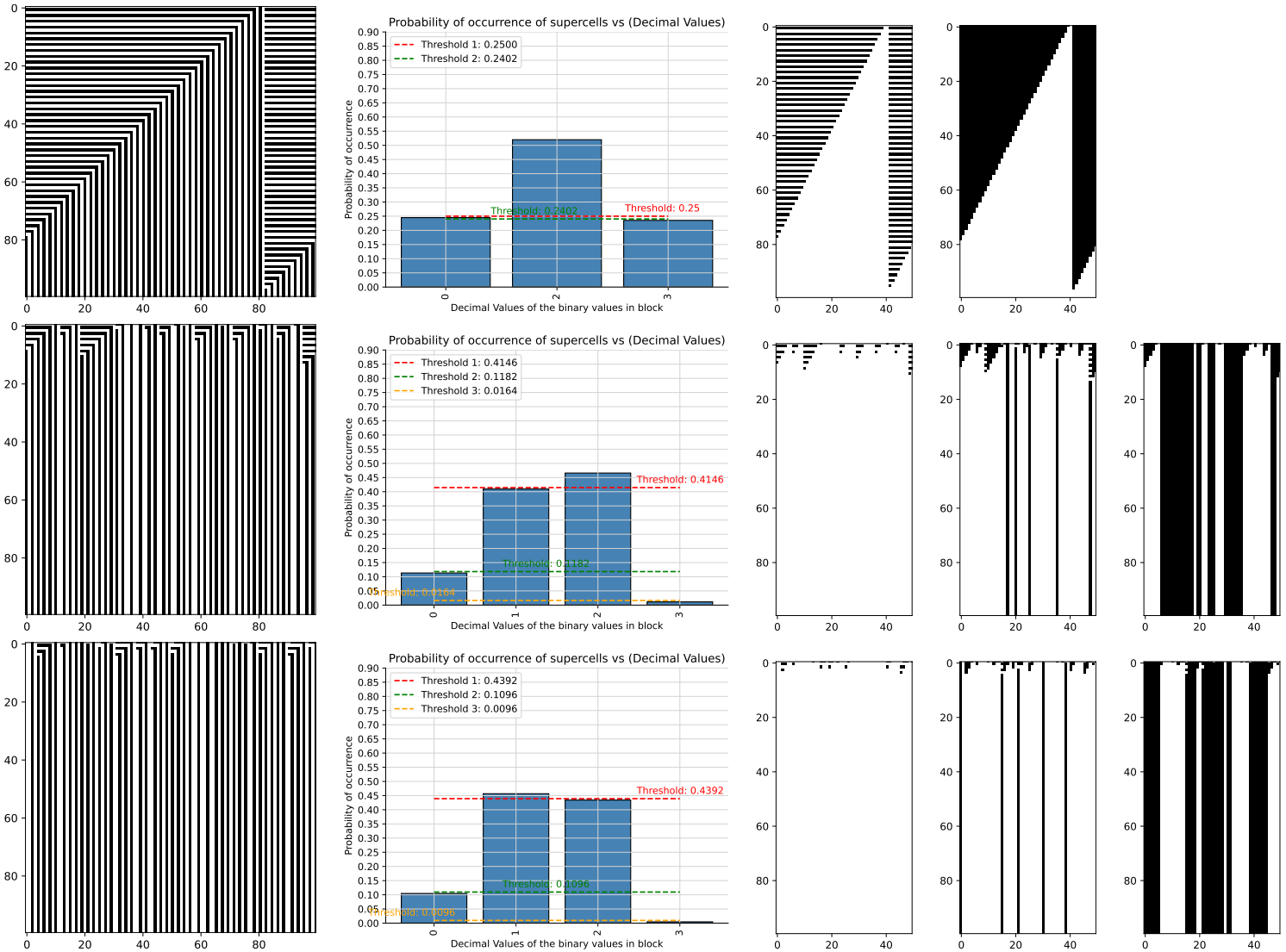

Table 15: FHCG plots for ECA Rule 14.

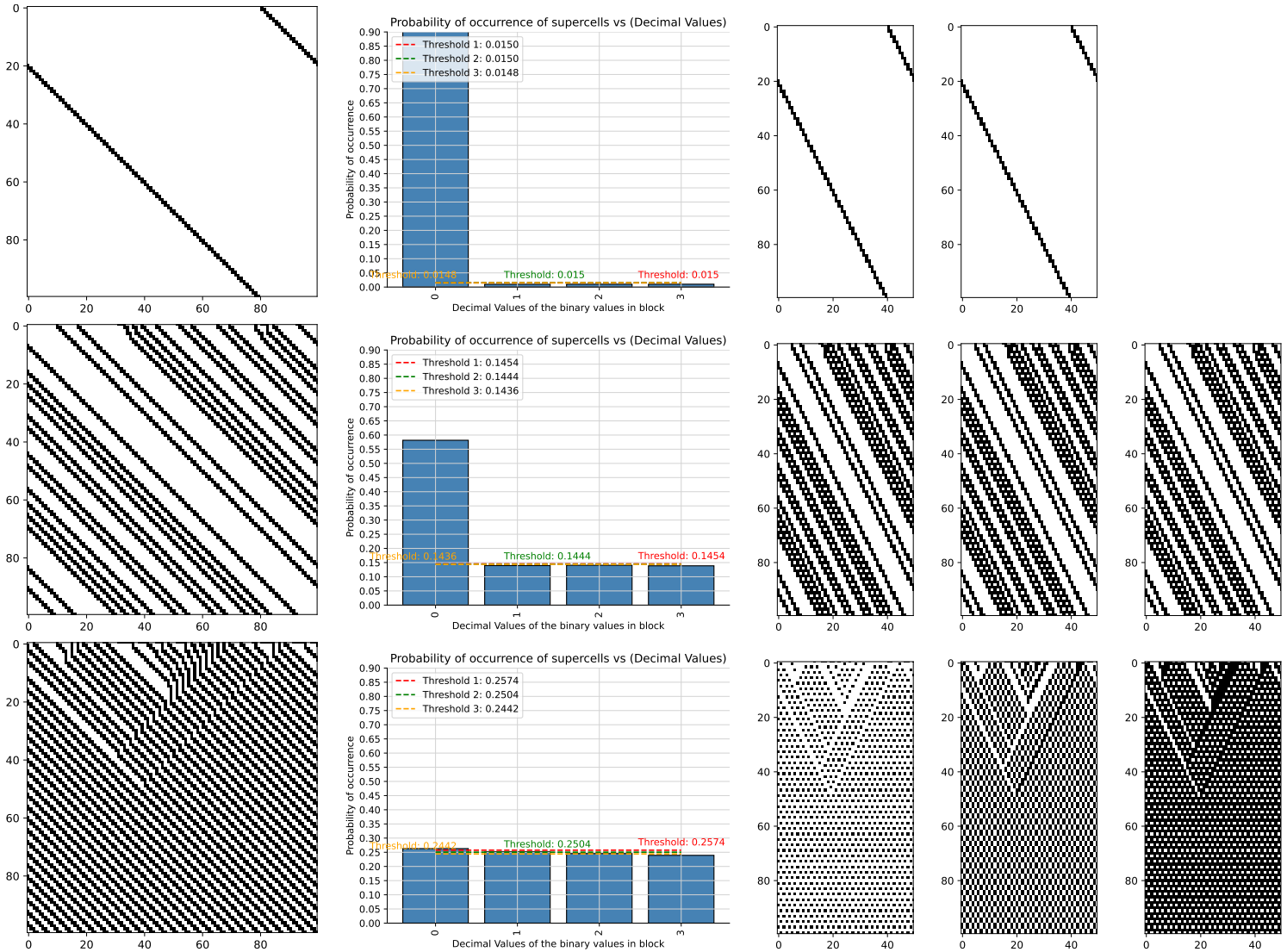

Table 16: FHCG plots for ECA Rule 15.

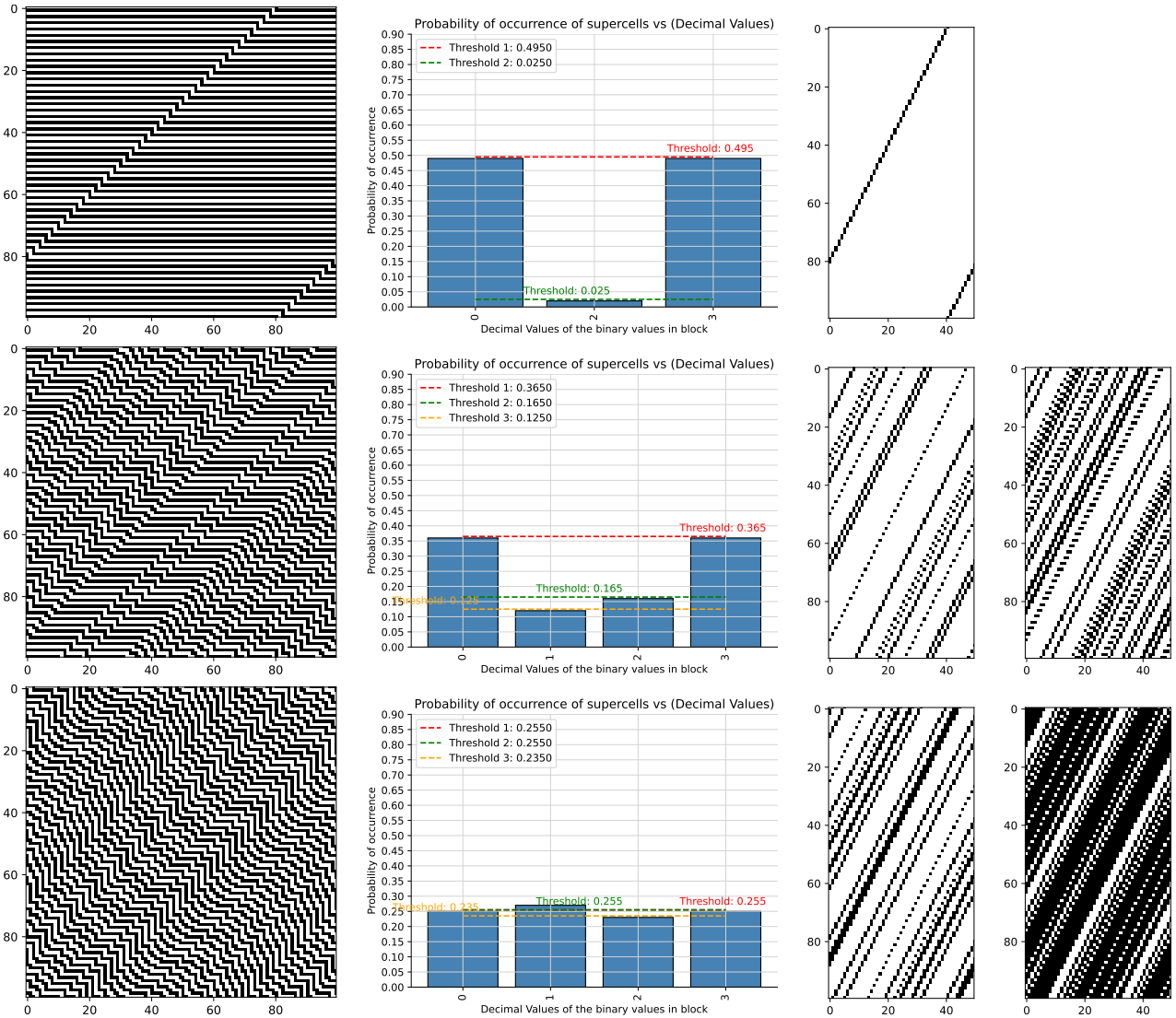

Table 17: FHCG plots for ECA Rule 18.

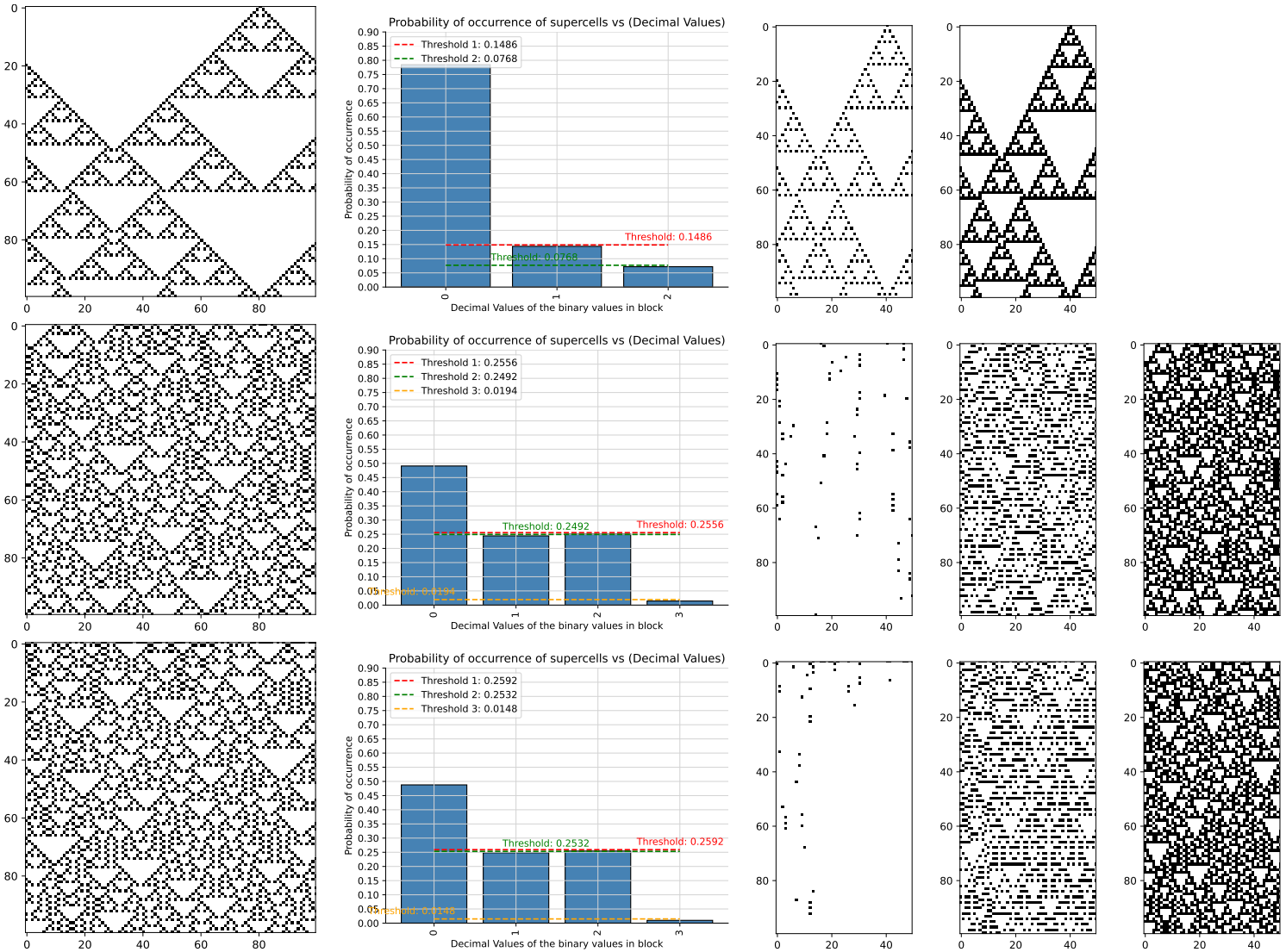

Table 18: FHCG plots for ECA Rule 19.

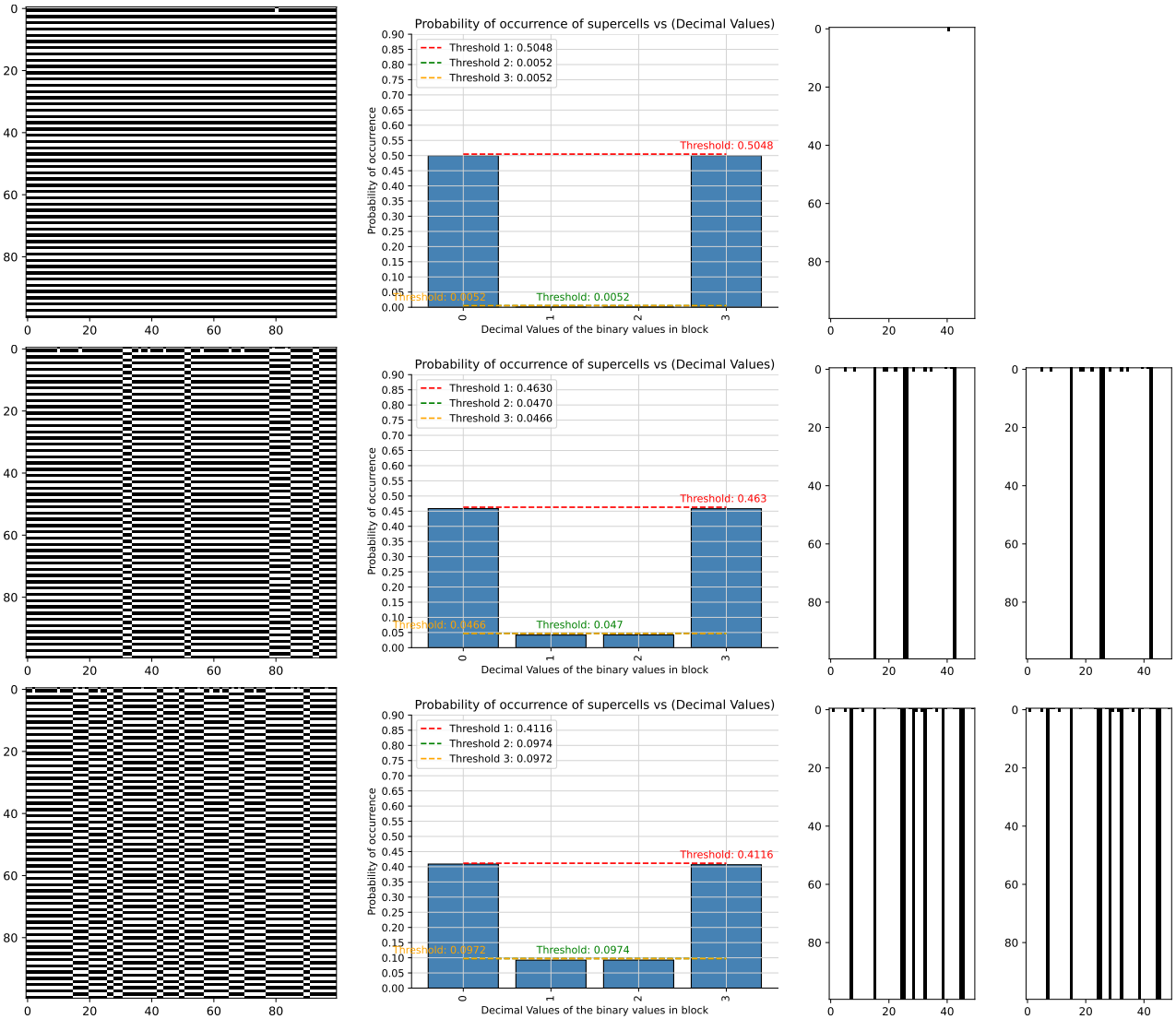

Table 19: FHCG plots for ECA Rule 22.

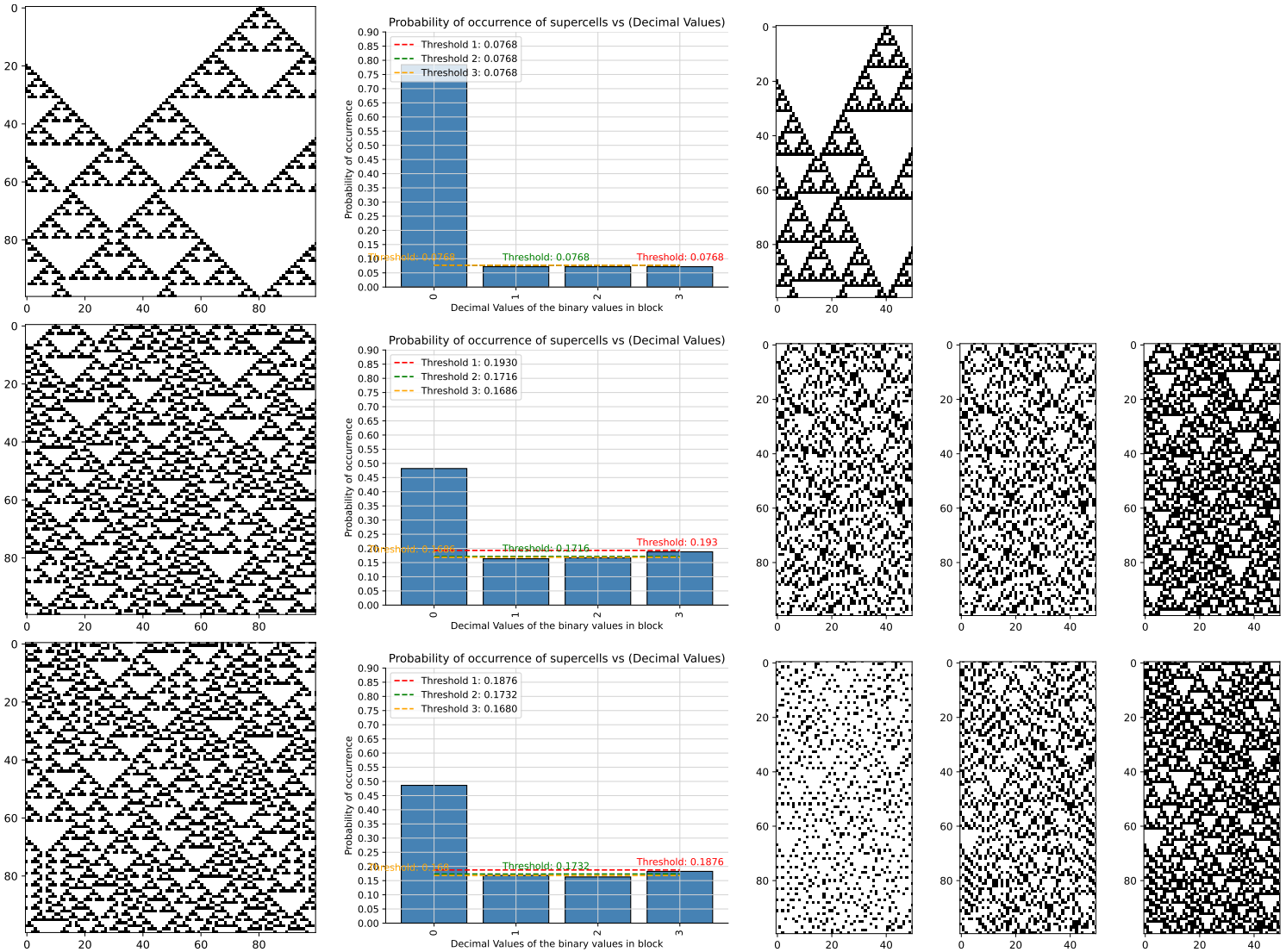

Table 20: FHCG plots for ECA Rule 23.

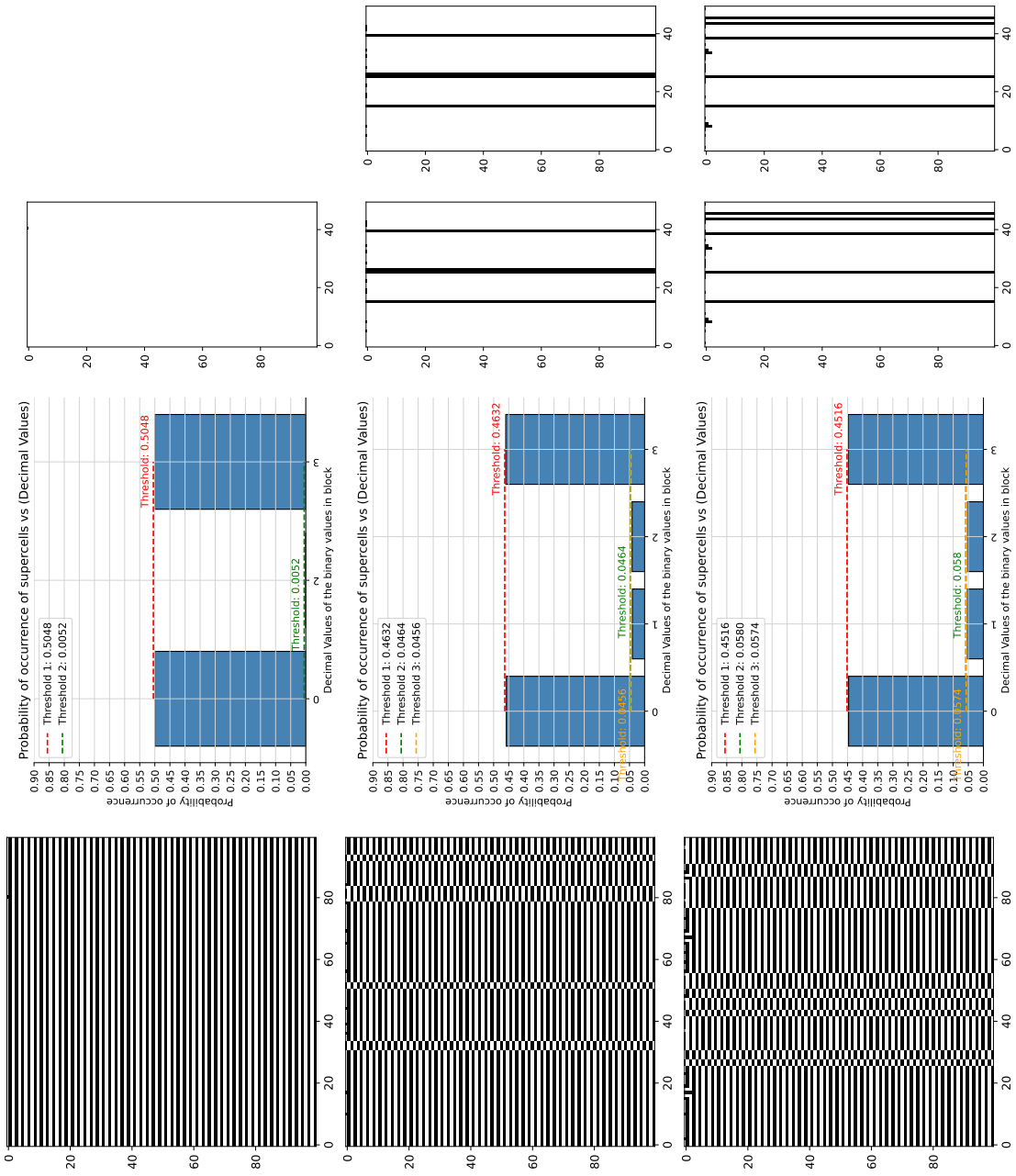

Table 21: FHCG plots for ECA Rule 24.

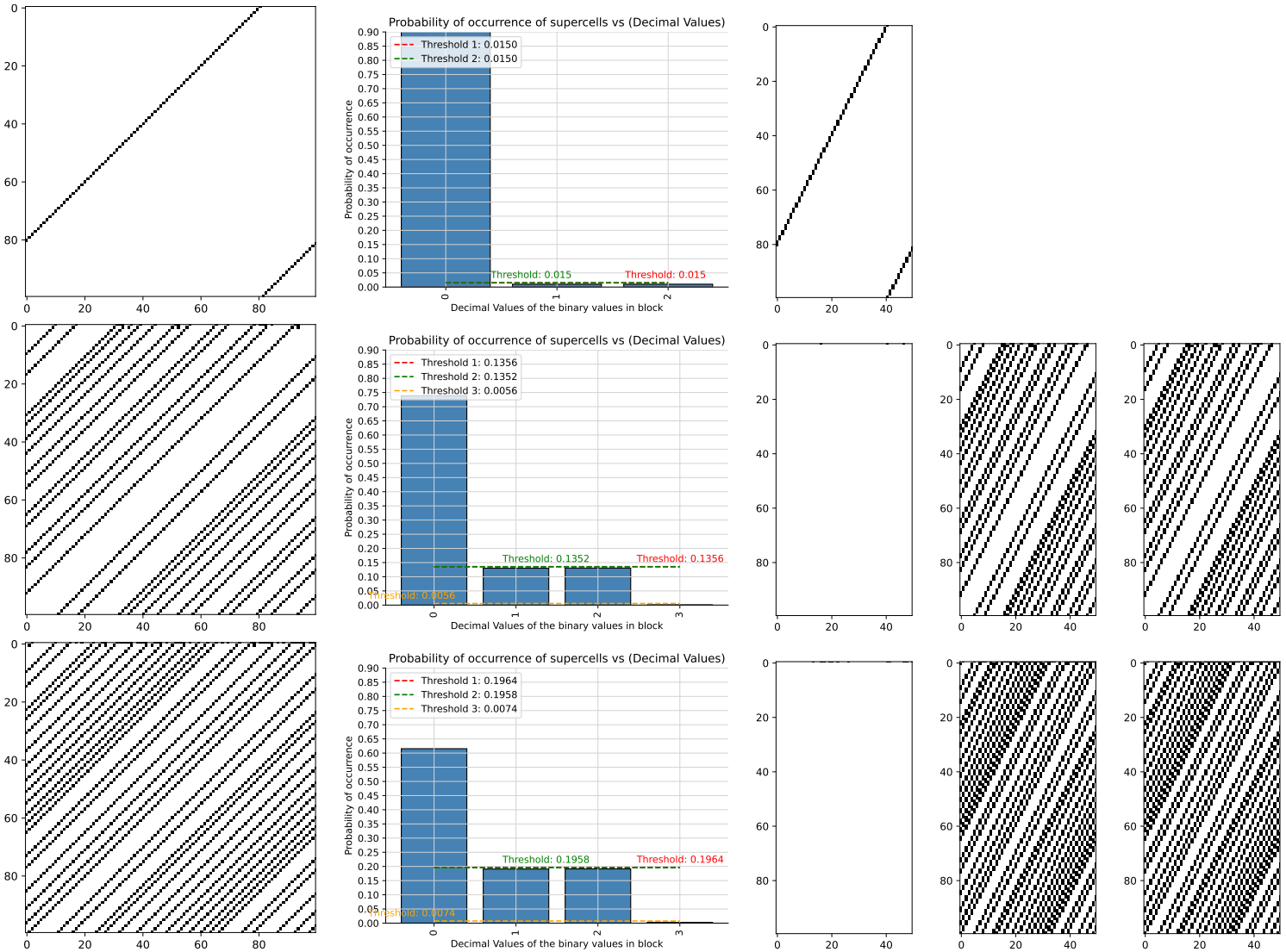

Table 22: FHCG plots for ECA Rule 25.

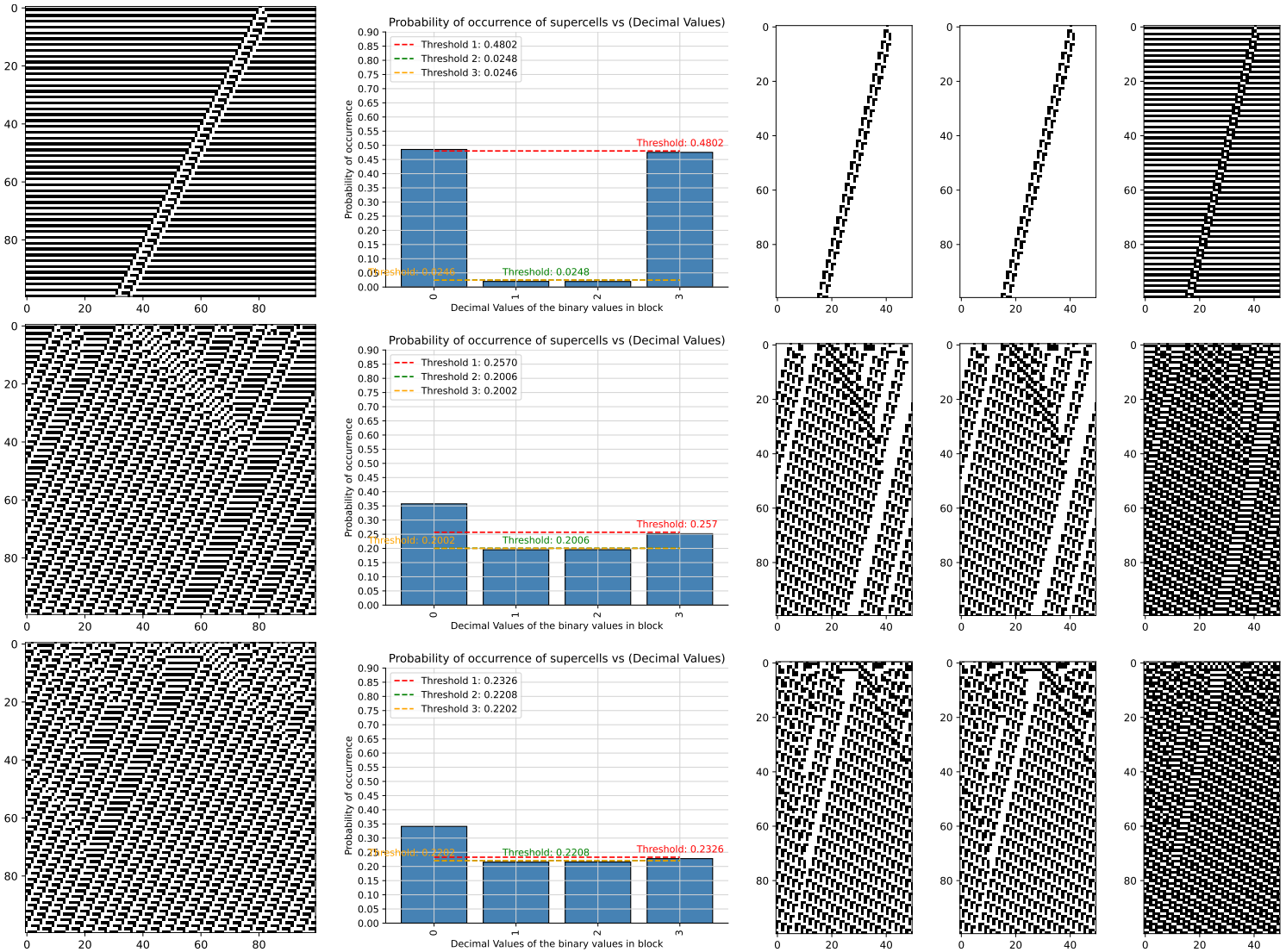

Table 23: FHCG plots for ECA Rule 26.

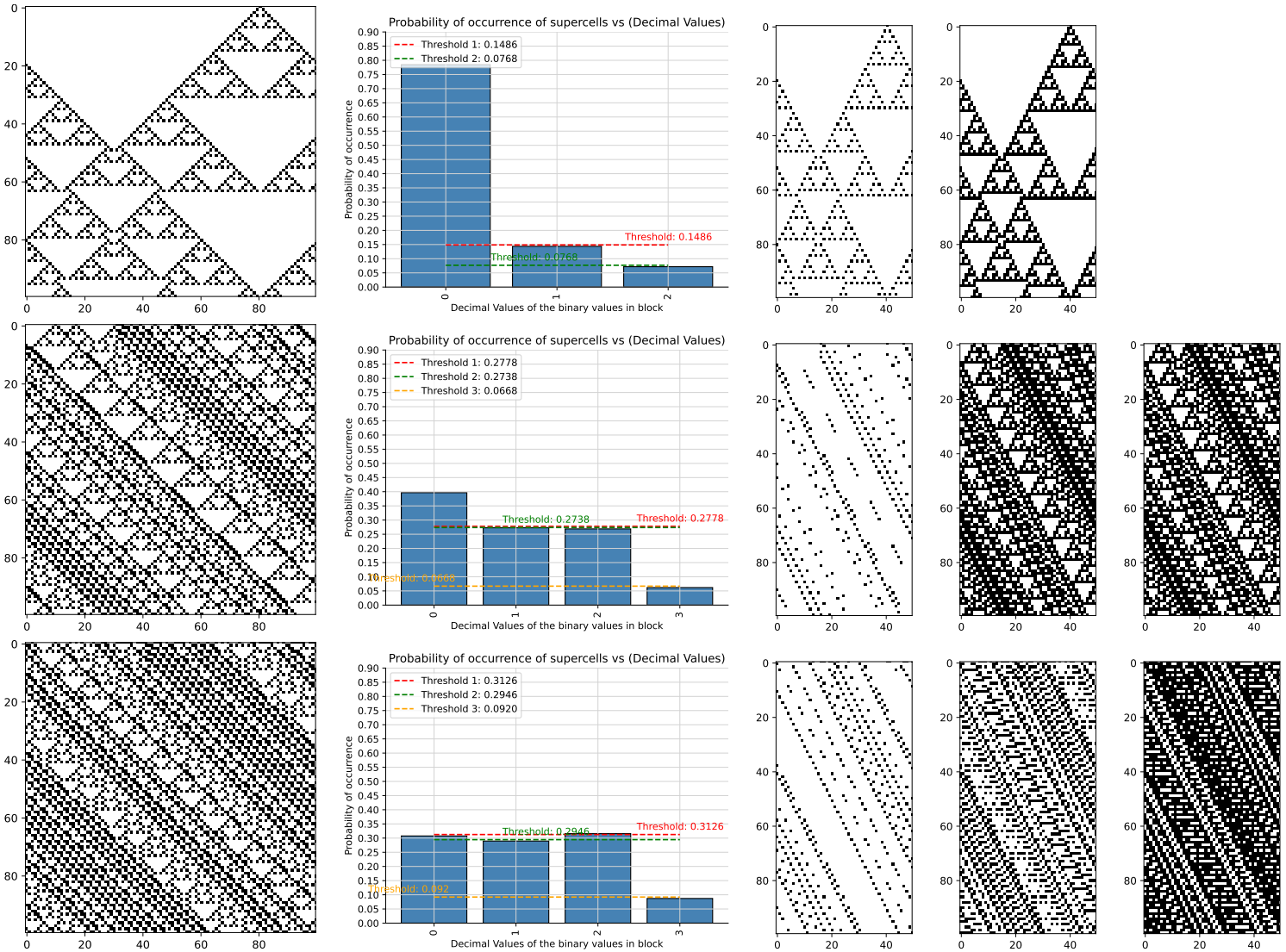

Table 24: FHCG plots for ECA Rule 27.

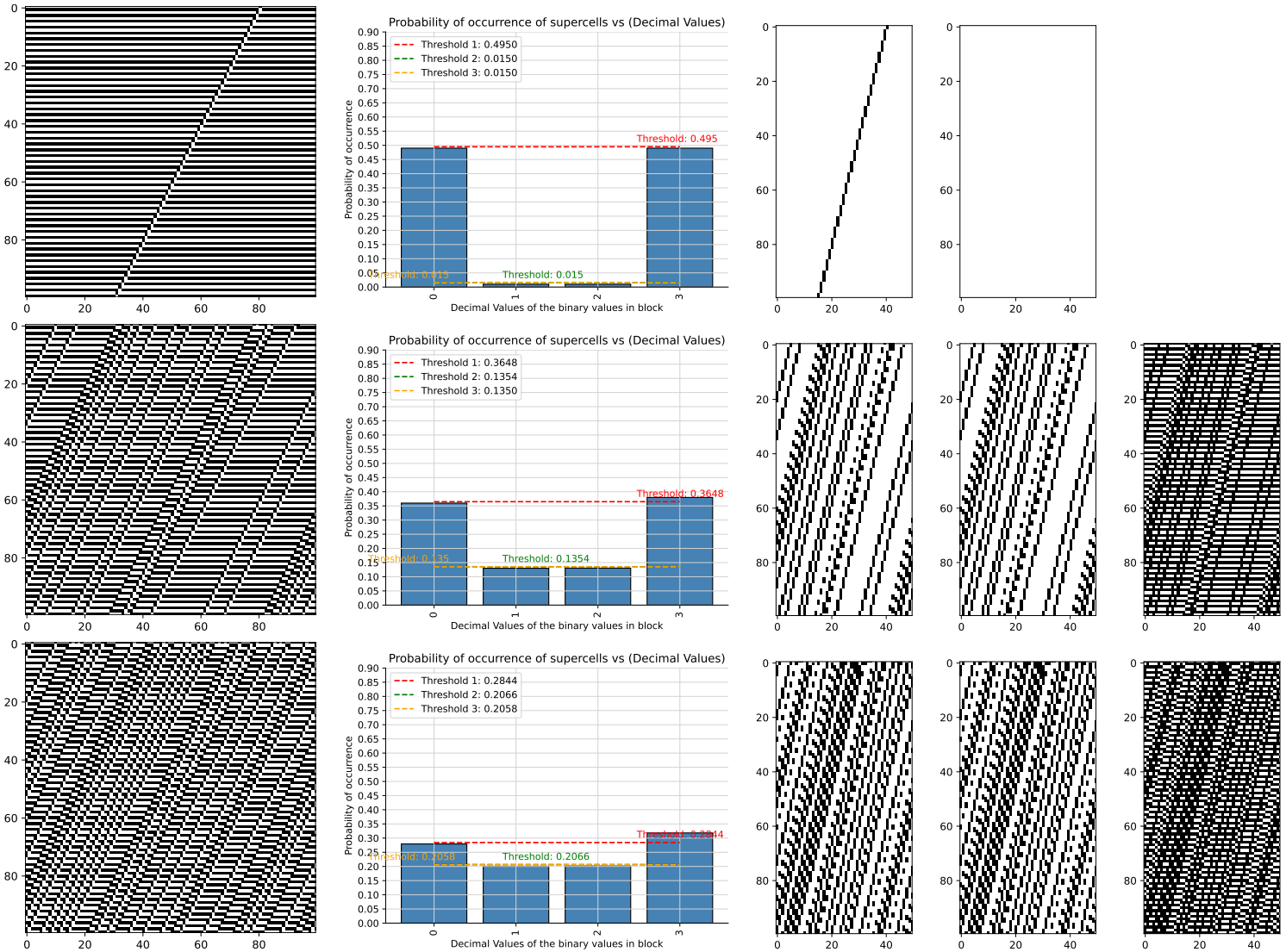

Table 25: FHCG plots for ECA Rule 28.

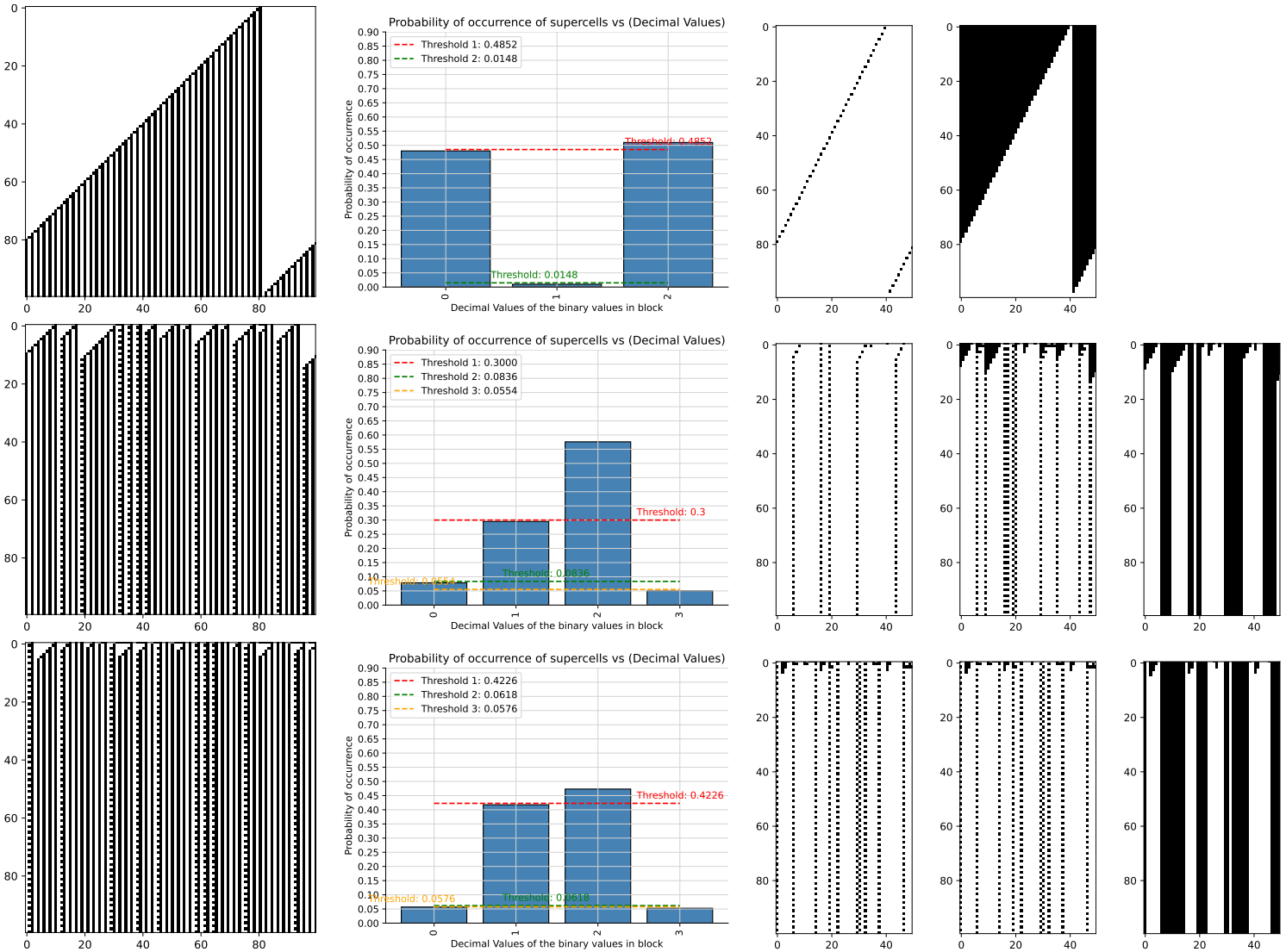

Table 26: FHCG plots for ECA Rule 29.

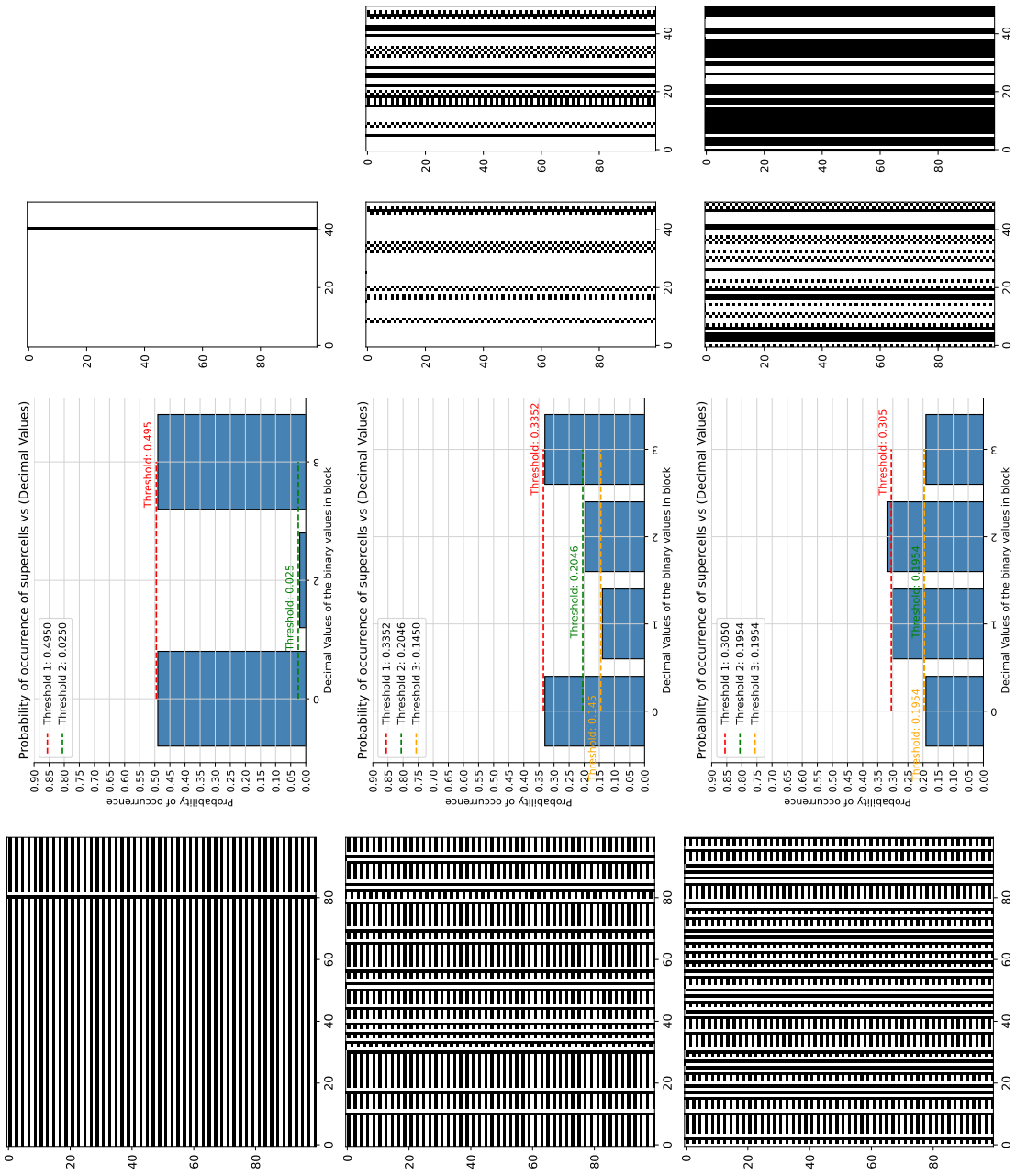

Table 27: FHCG plots for ECA Rule 30.

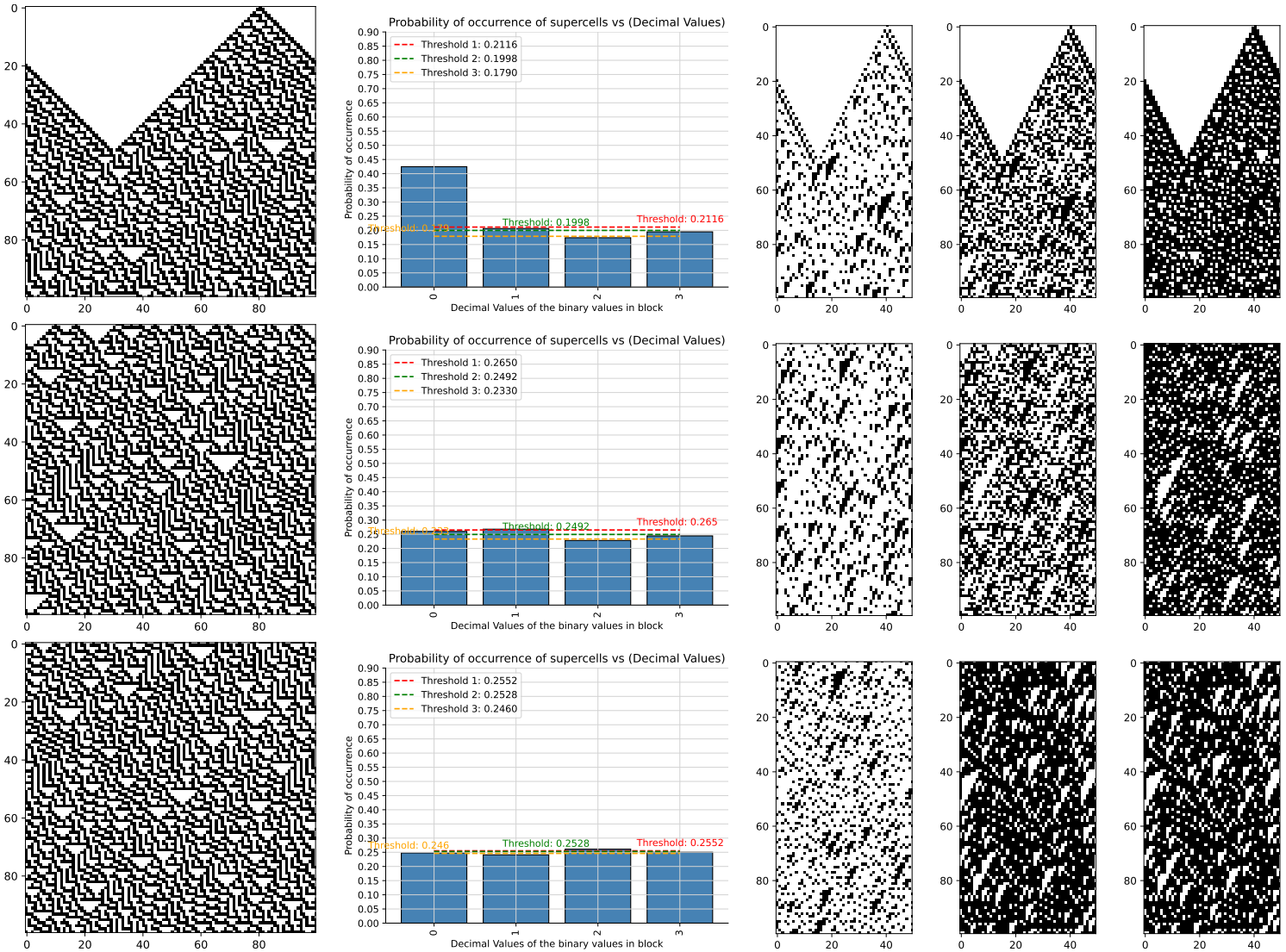

Table 28: FHCG plots for ECA Rule 32.

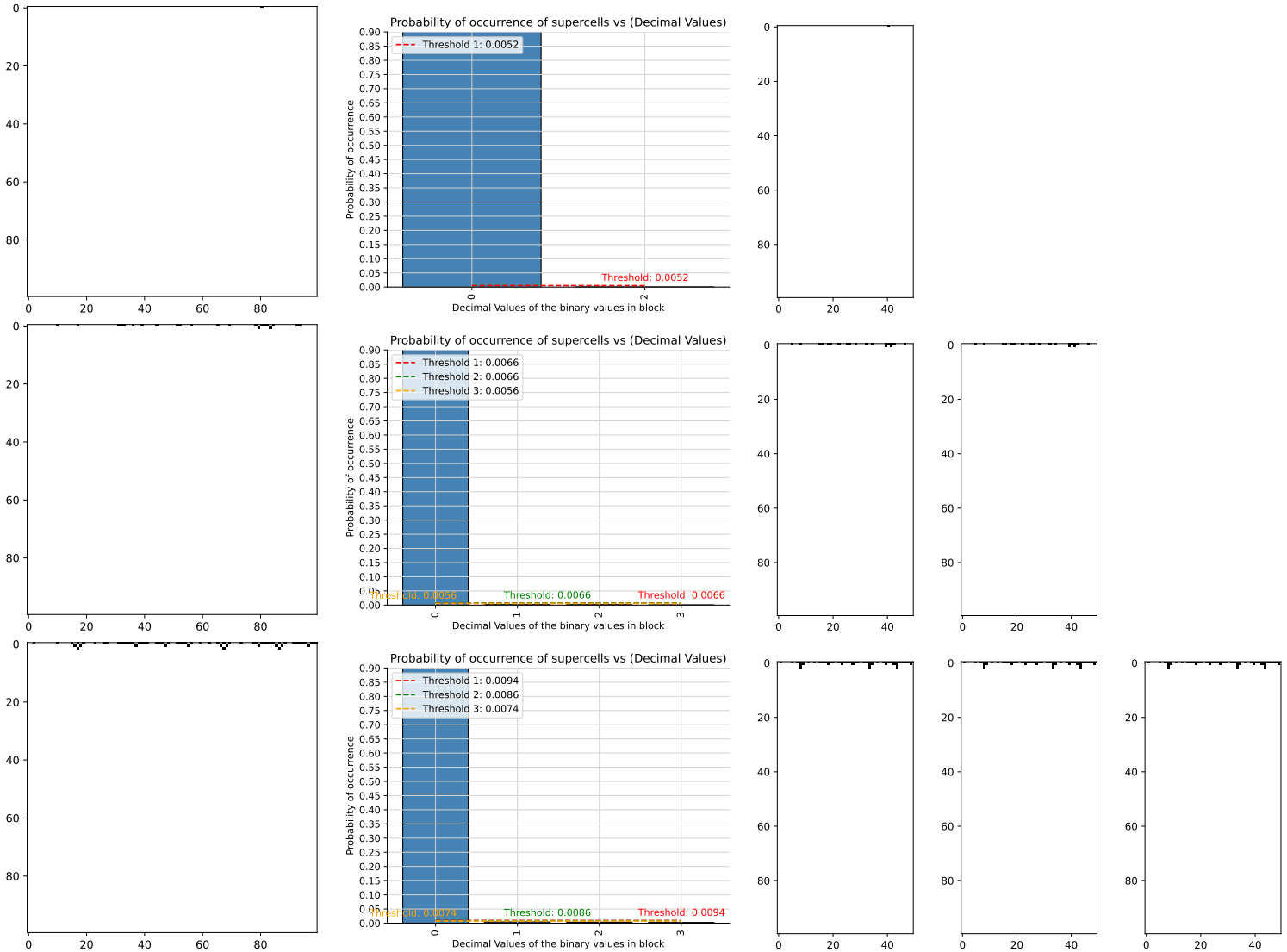

Table 29: FHCG plots for ECA Rule 33.

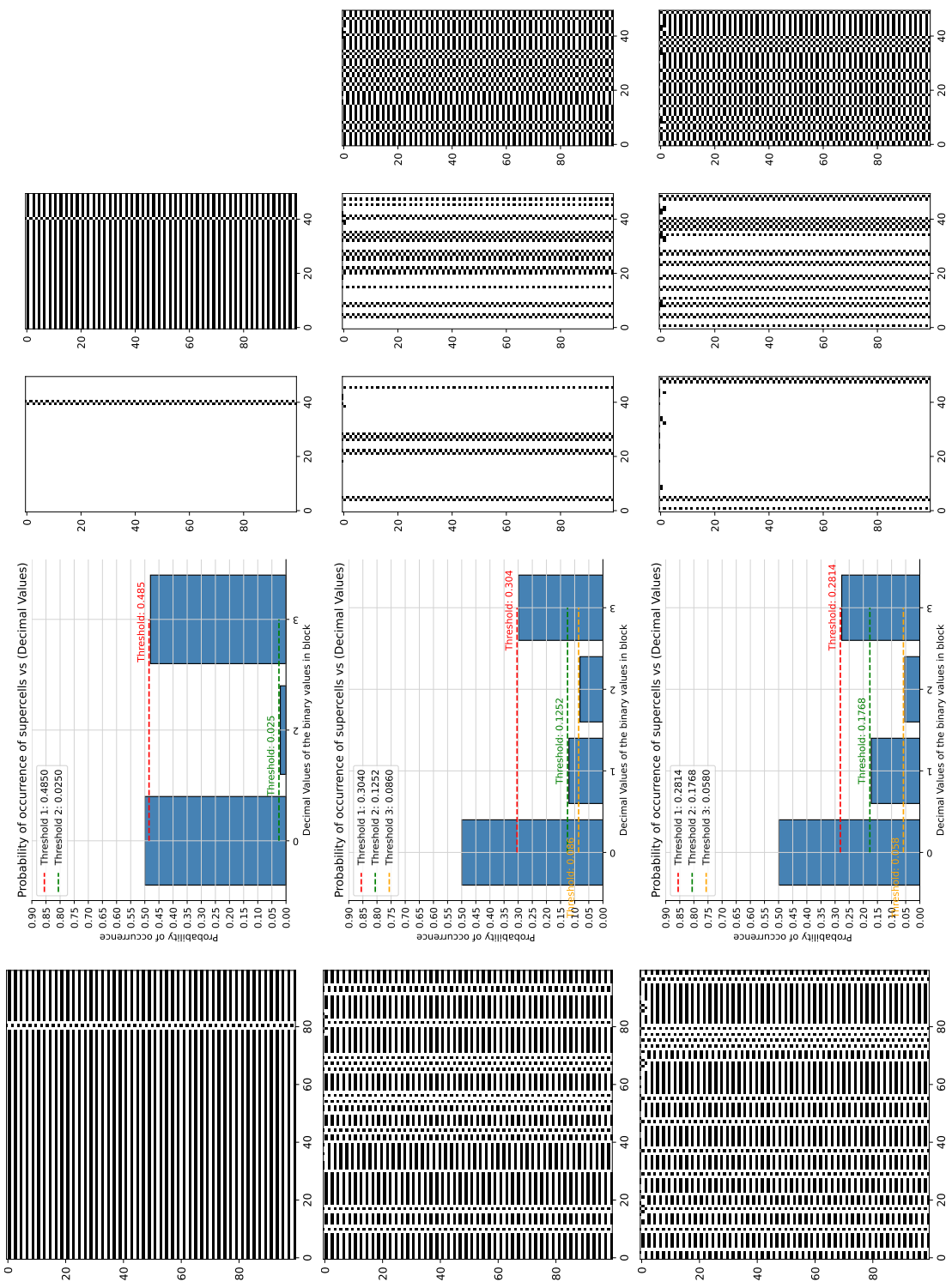

Table 30: FHCG plots for ECA Rule 34.

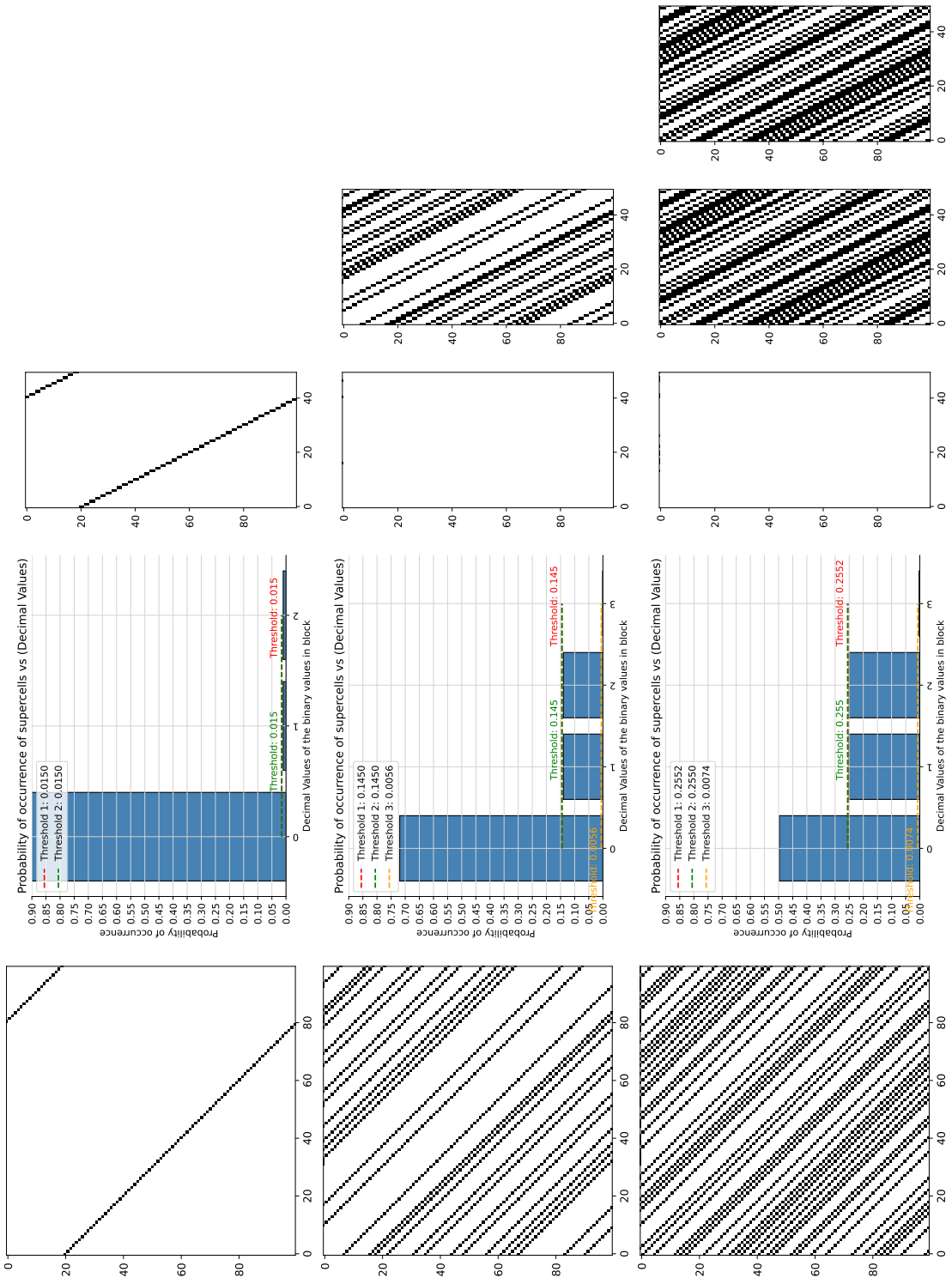

Table 31: FHCG plots for ECA Rule 35.

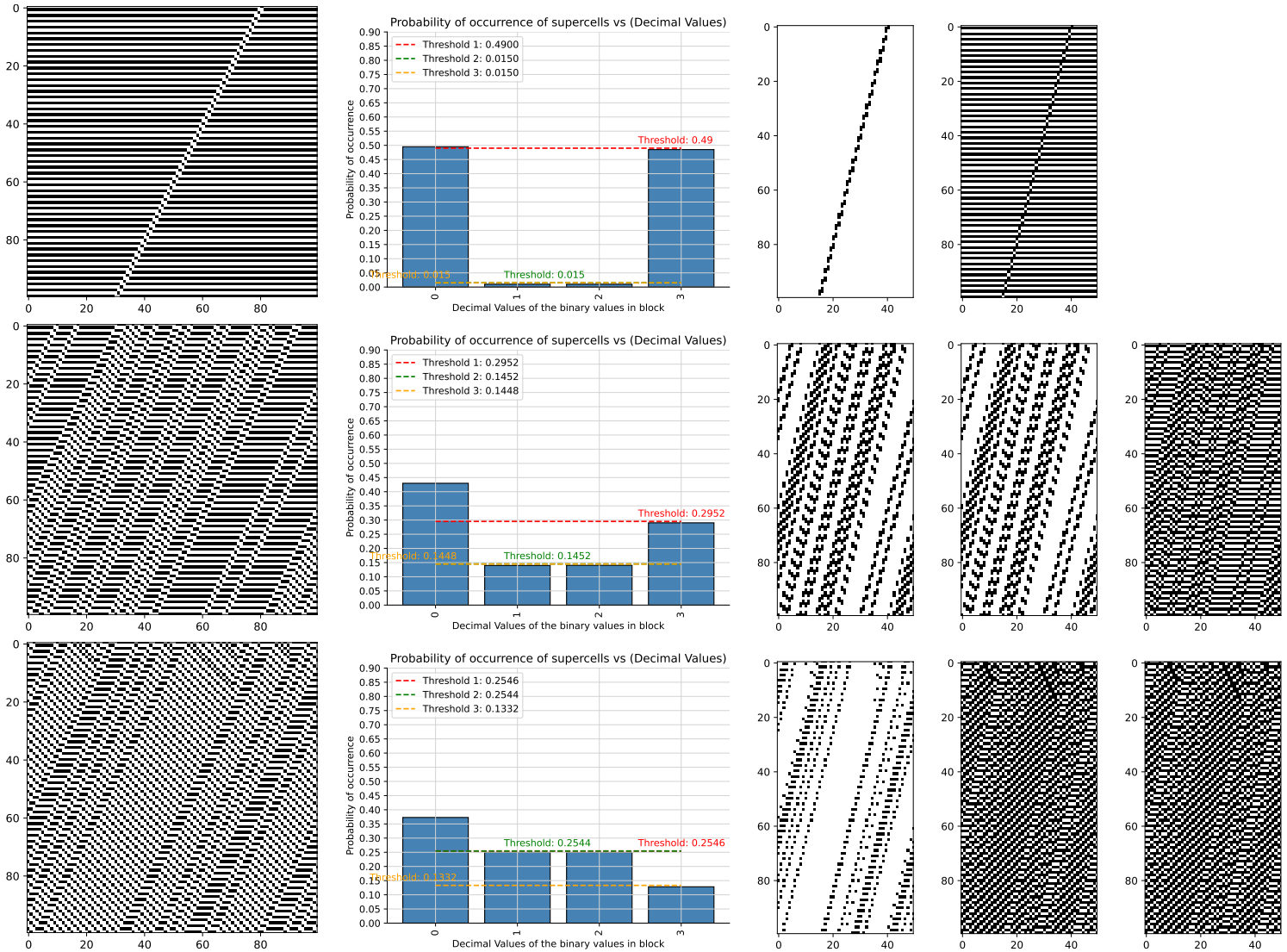

Table 32: FHCG plots for ECA Rule 36.

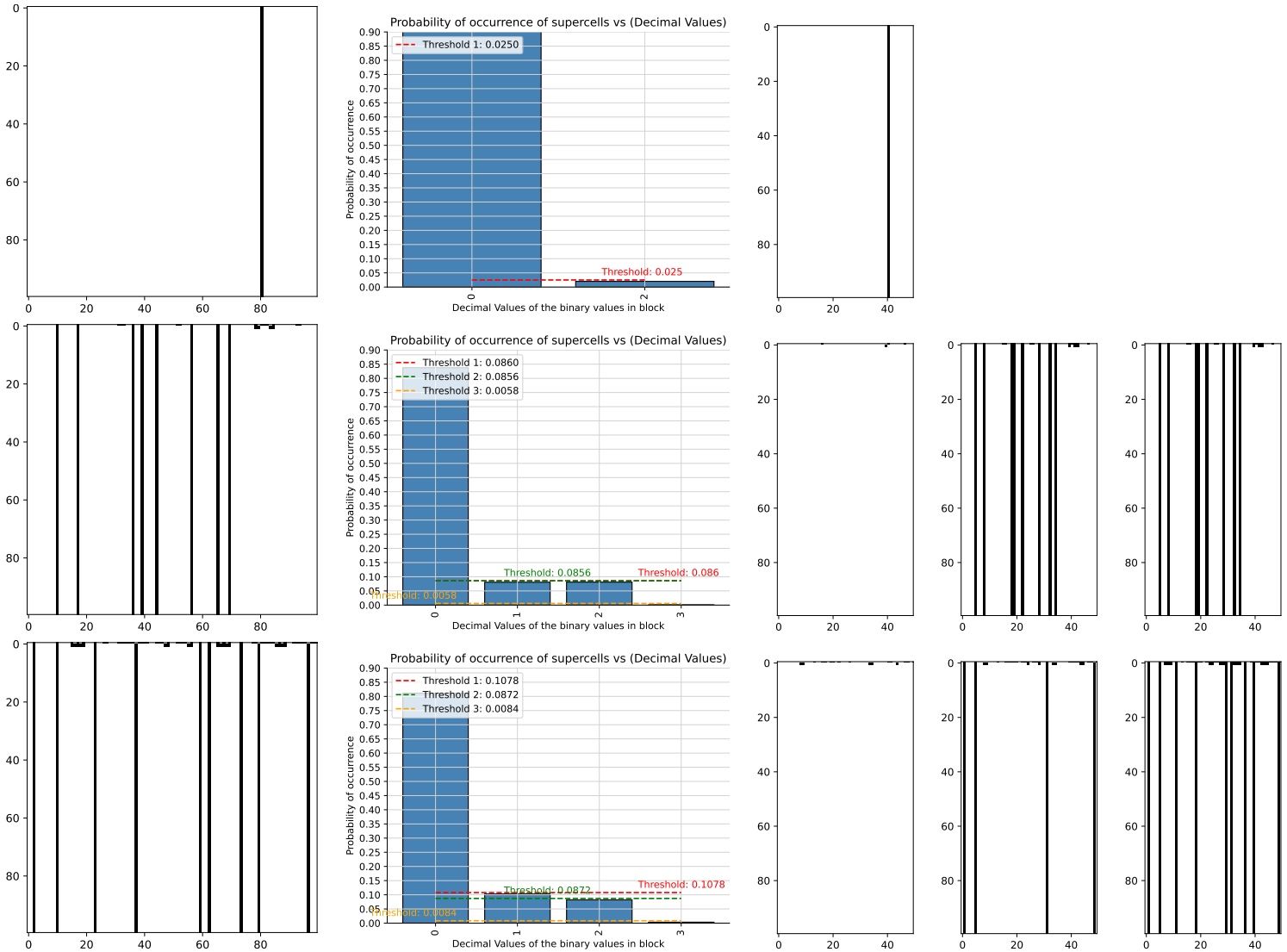

Table 33: FHCG plots for ECA Rule 37.

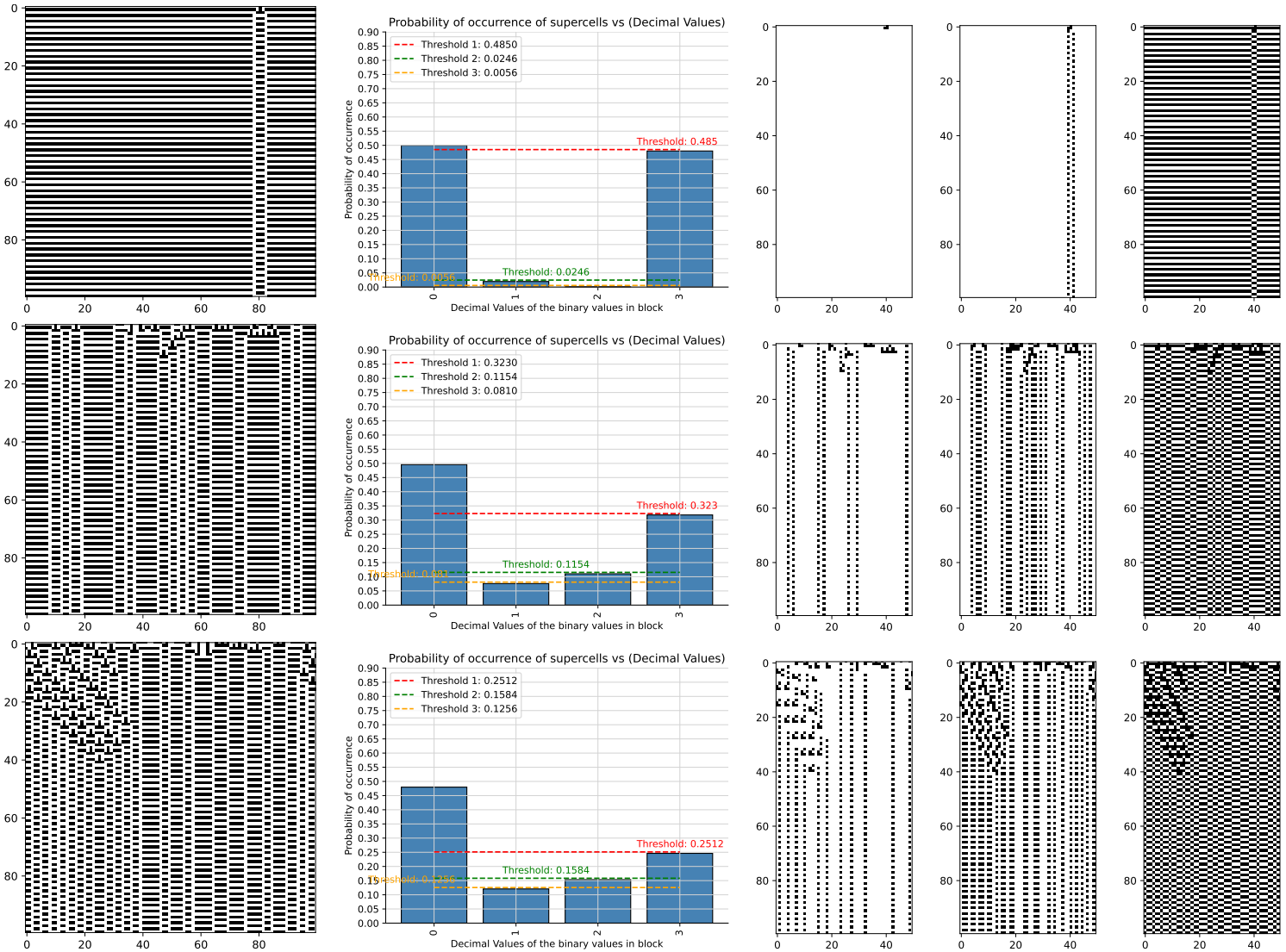

Table 34: FHCG plots for ECA Rule 38.

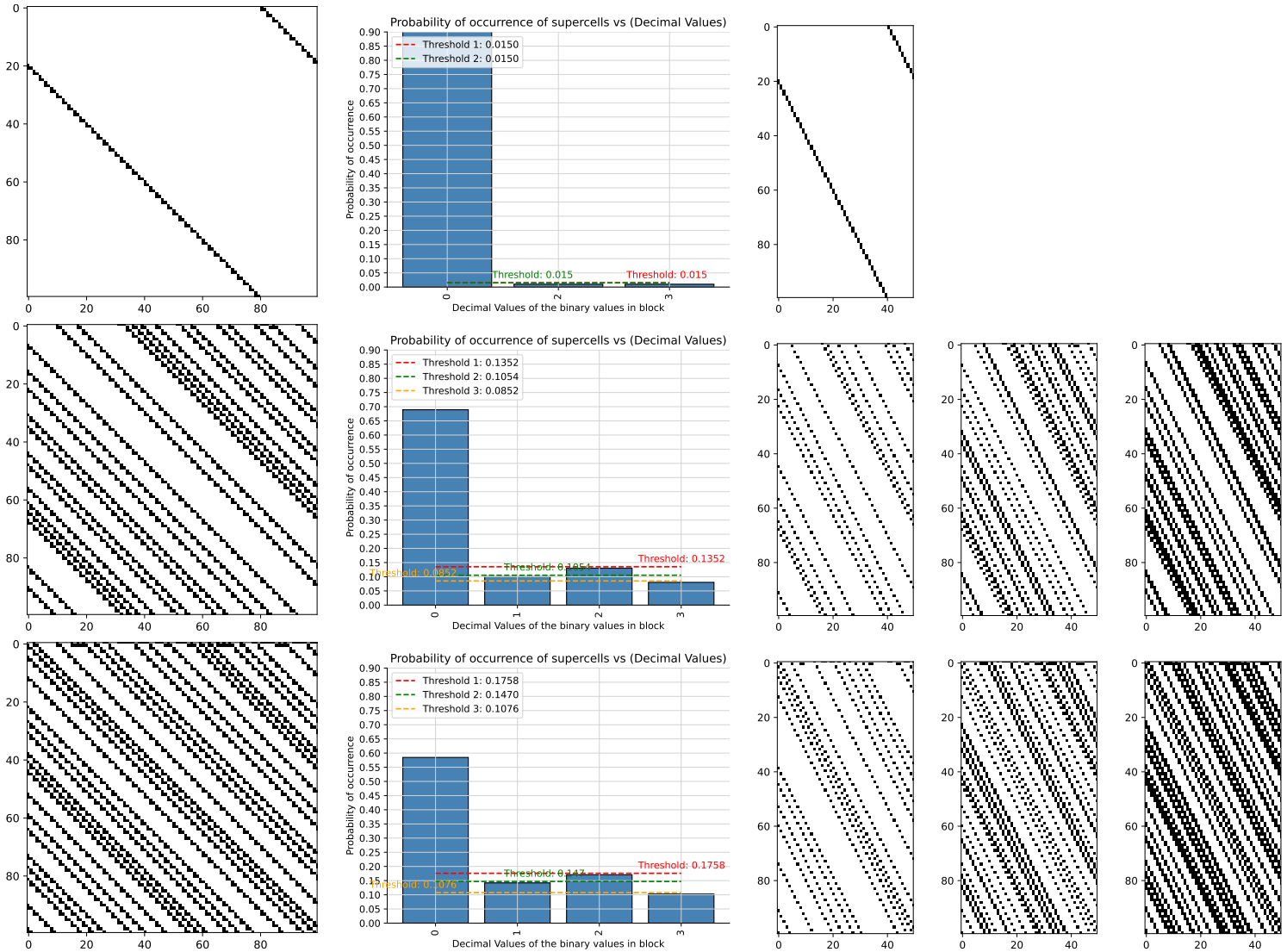

Table 35: FHCG plots for ECA Rule 40.

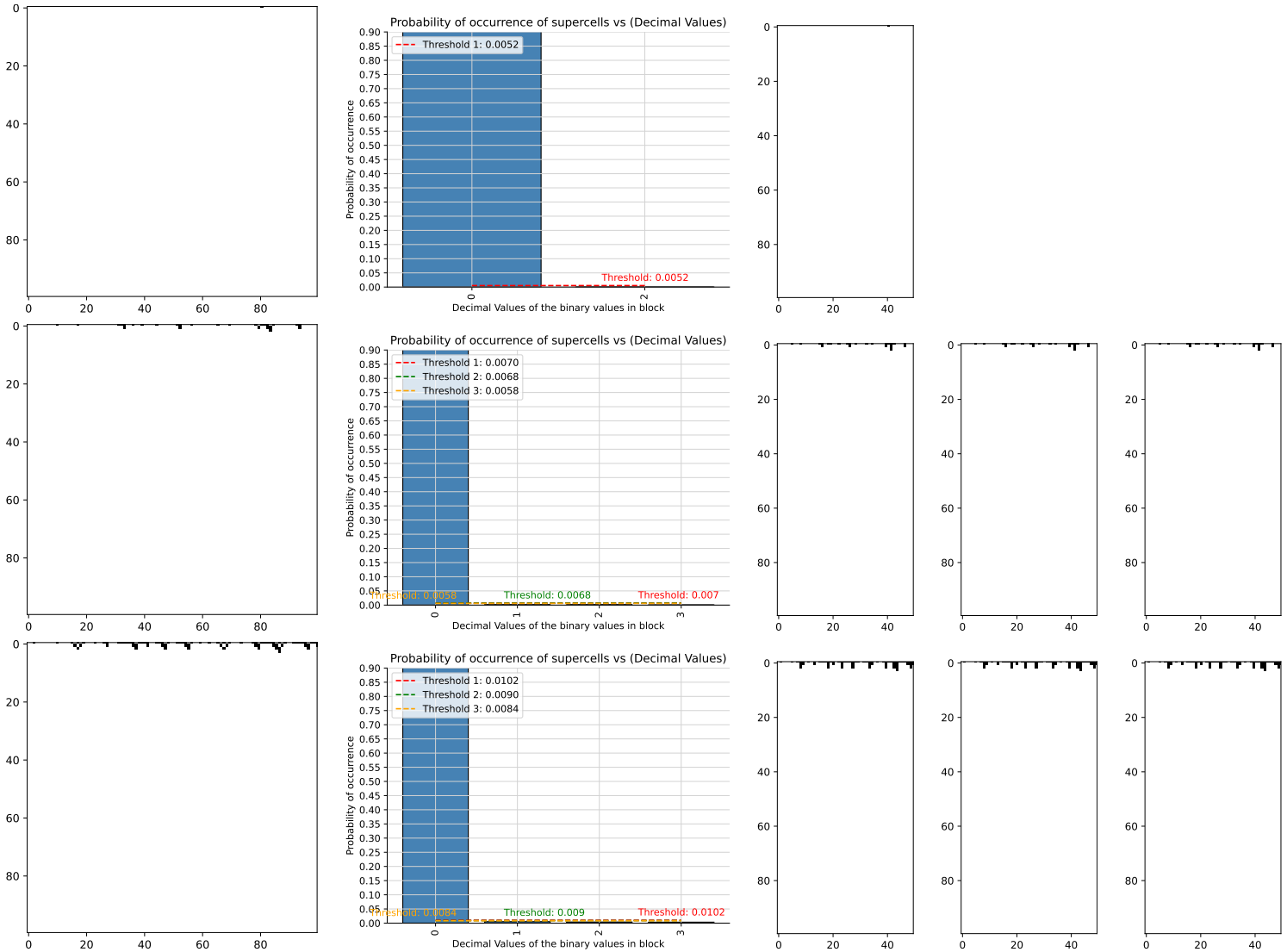

Table 36: FHCG plots for ECA Rule 41.

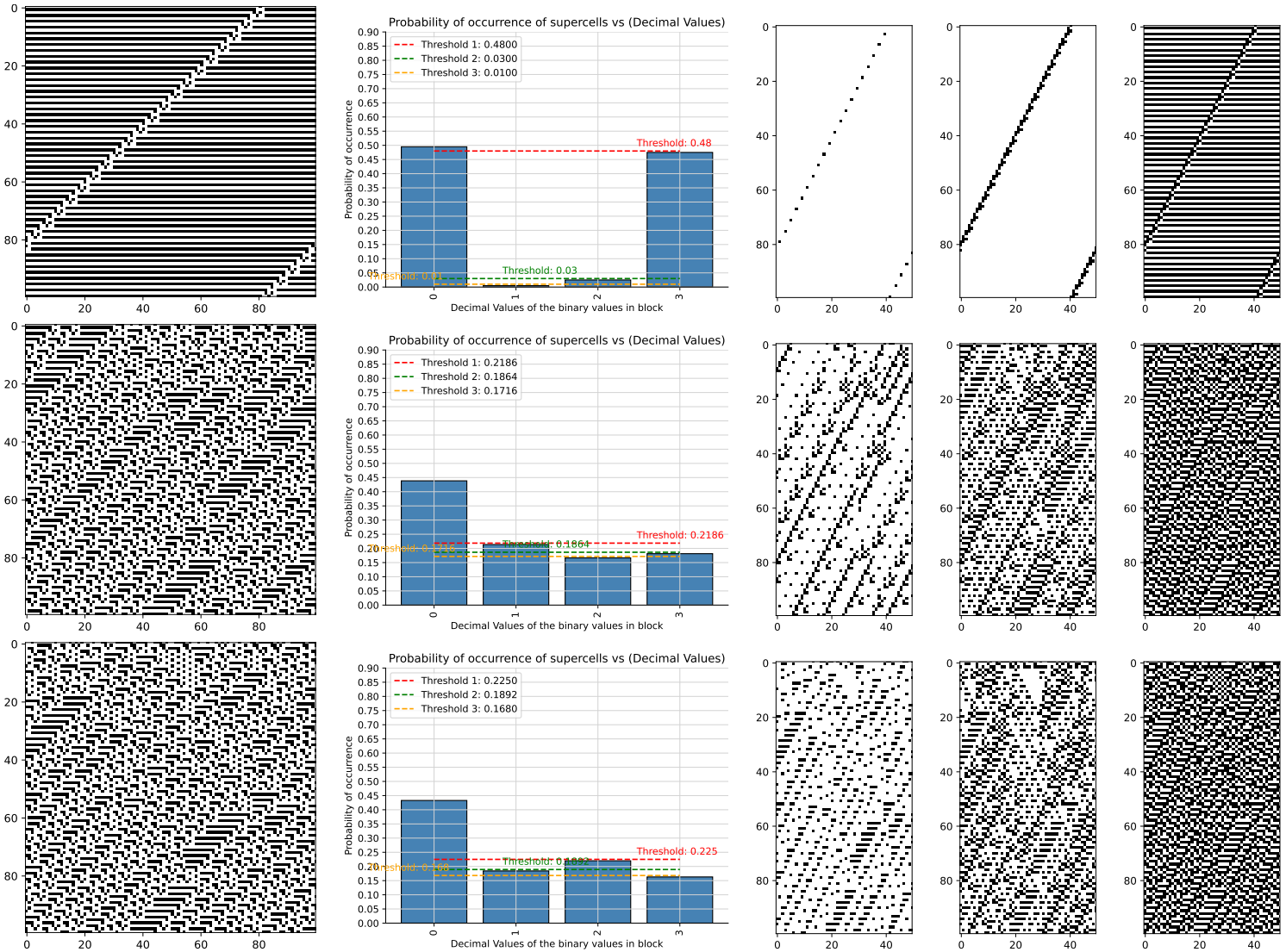

Table 37: FHCG plots for ECA Rule 42.

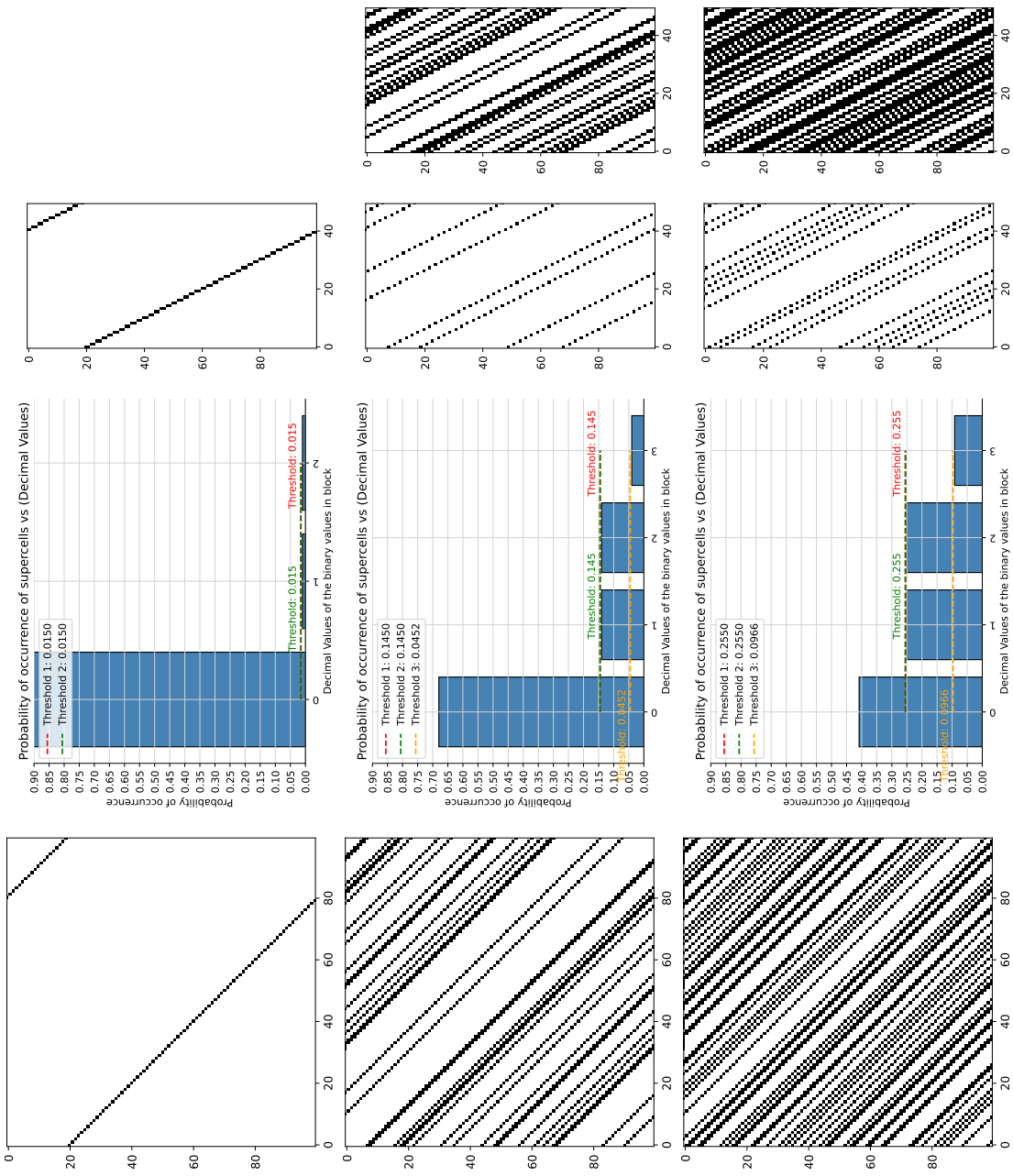

Table 38: FHCG plots for ECA Rule 43.

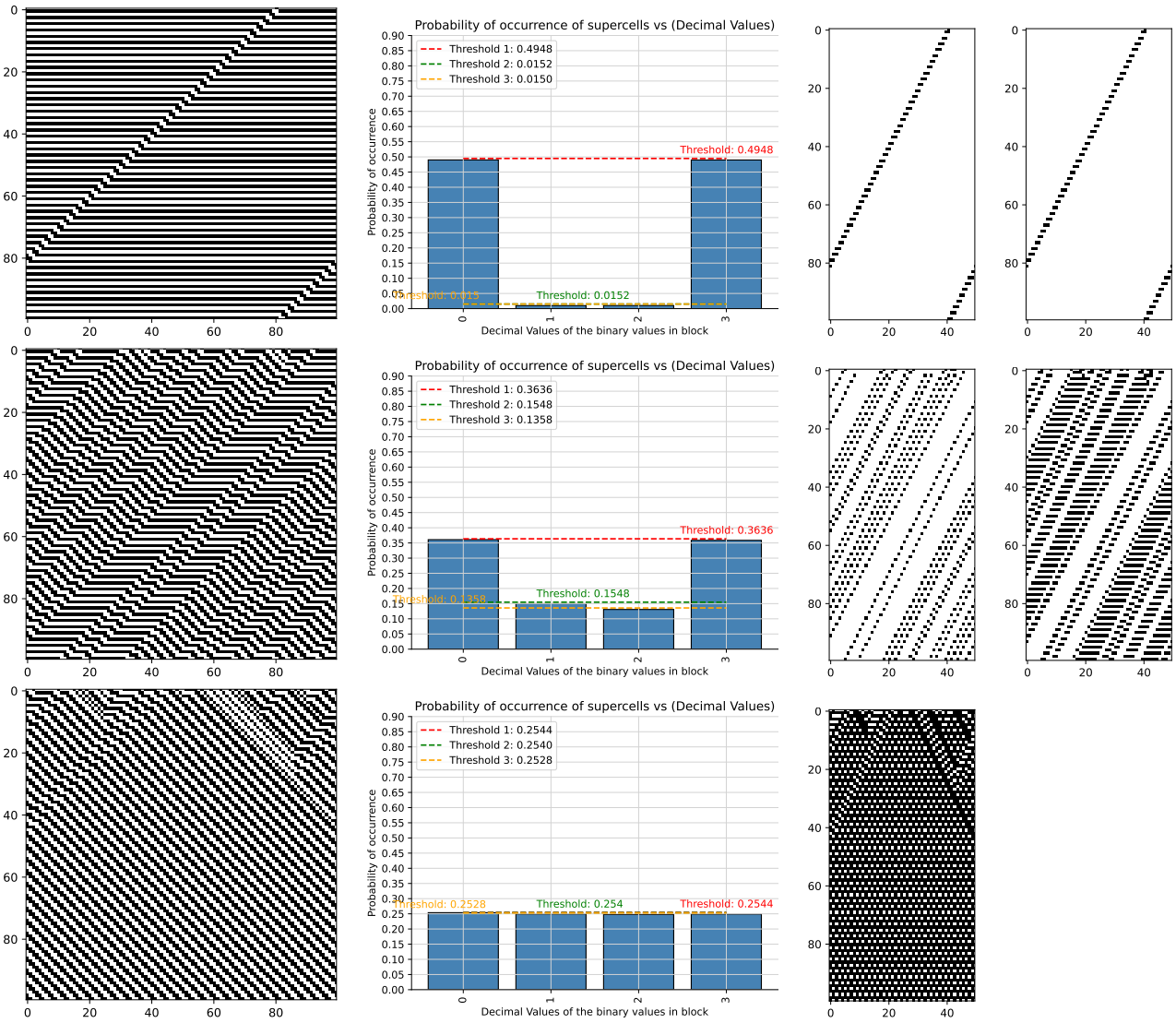

Table 39: FHCG plots for ECA Rule 44.

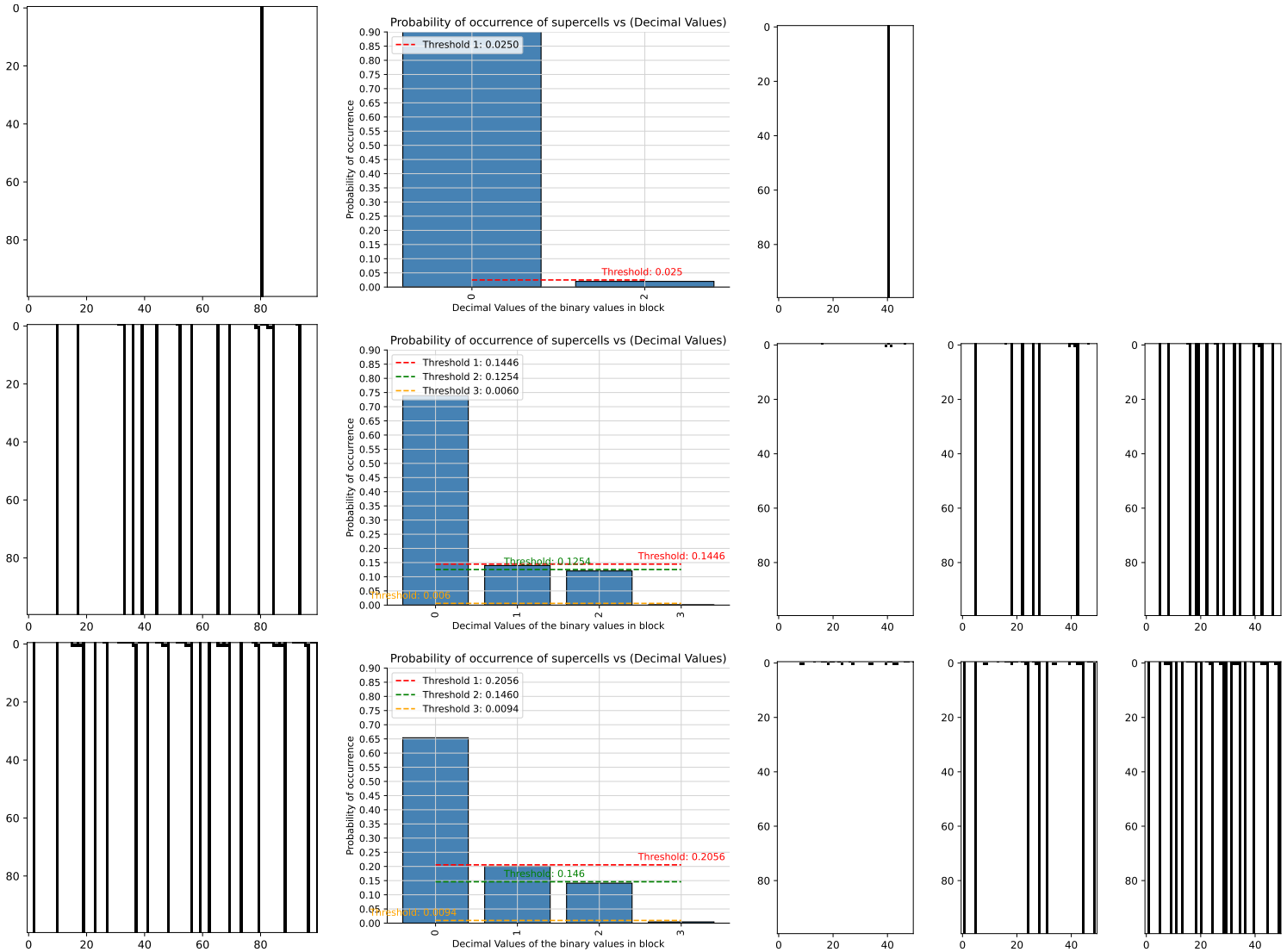

Table 40: FHCG plots for ECA Rule 45.

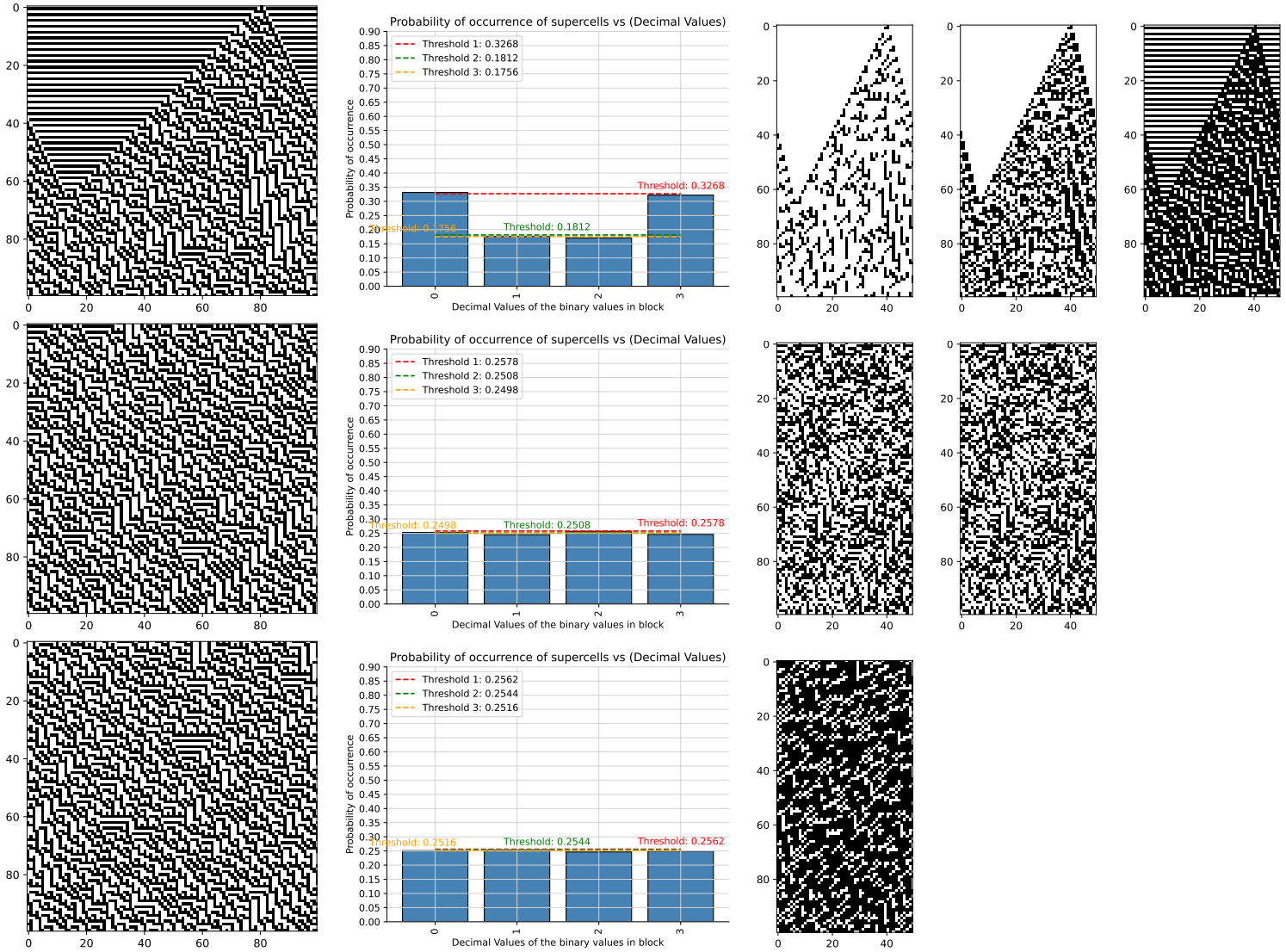

Table 41: FHCG plots for ECA Rule 46.

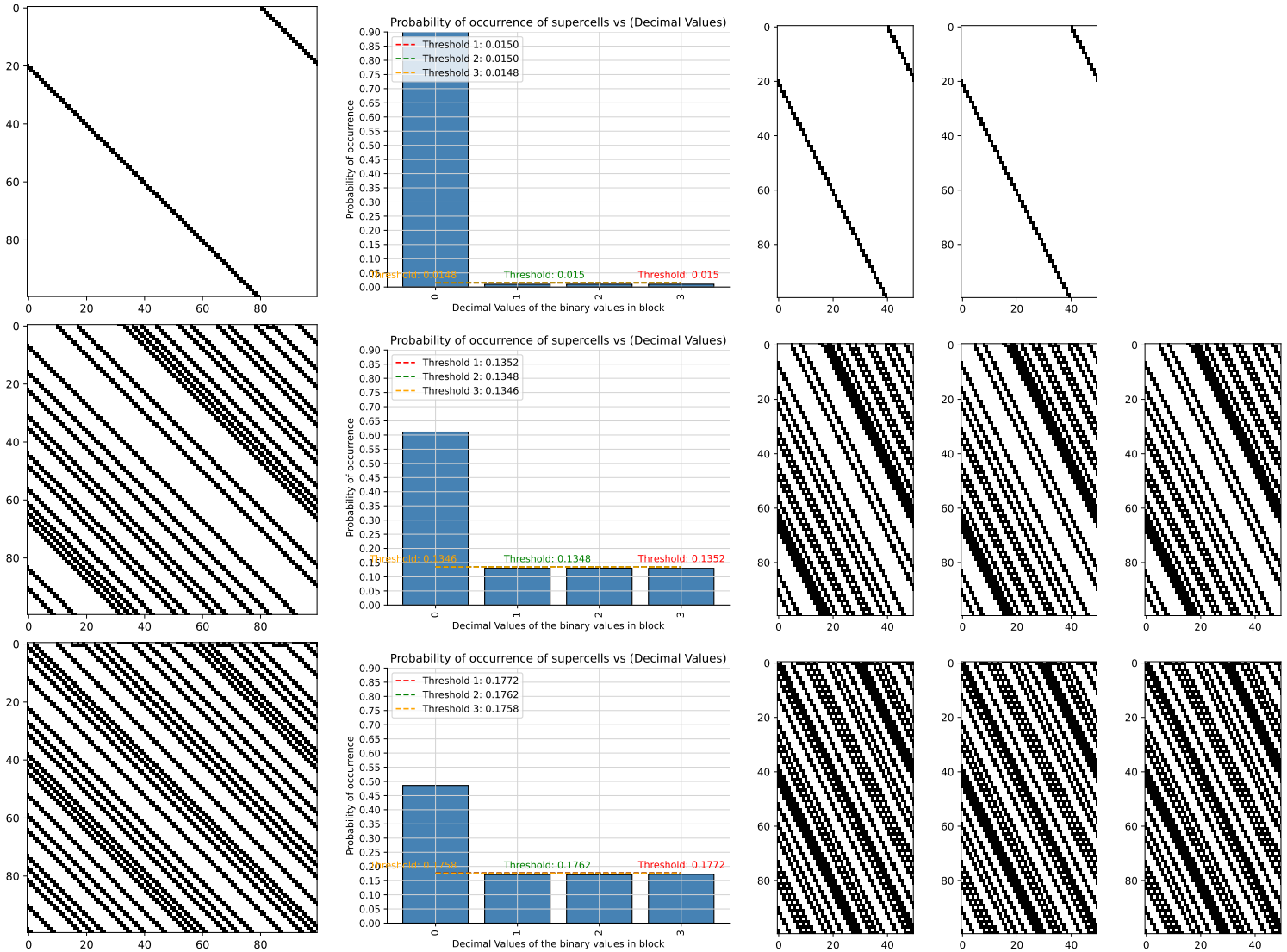

Table 42: FHCG plots for ECA Rule 50.

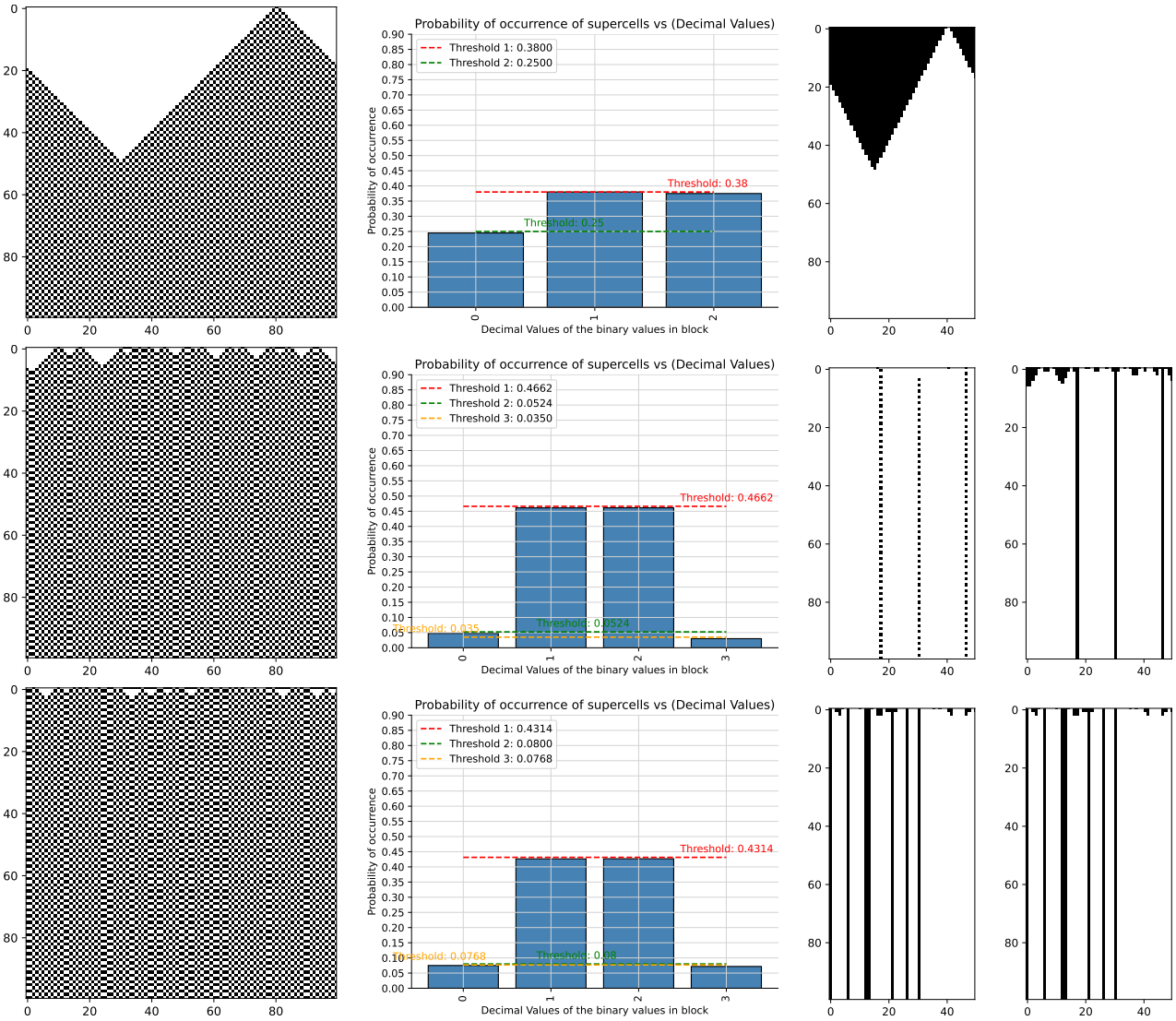

Table 43: FHCG plots for ECA Rule 51.

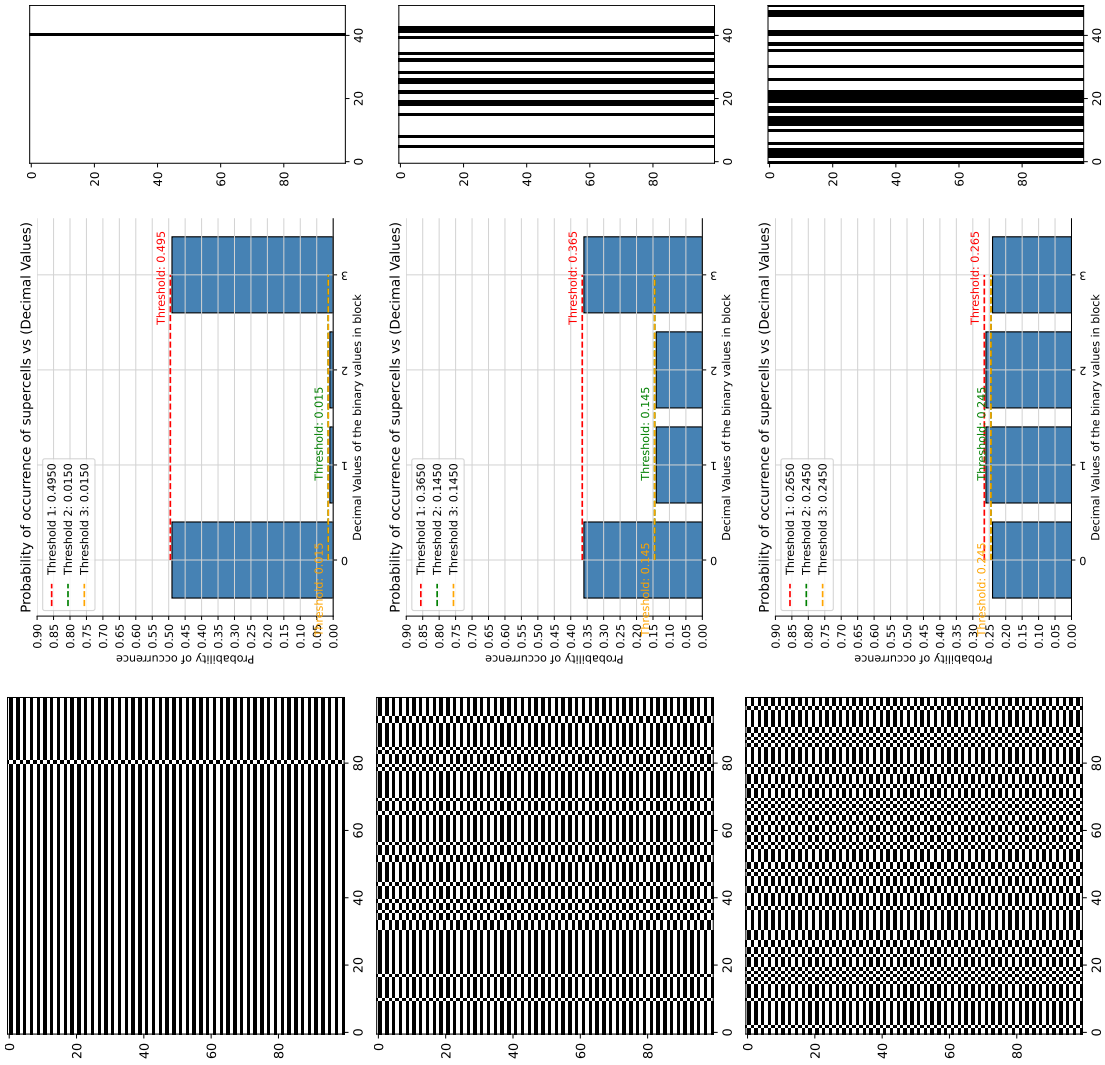

Table 44: FHCG plots for ECA Rule 54.

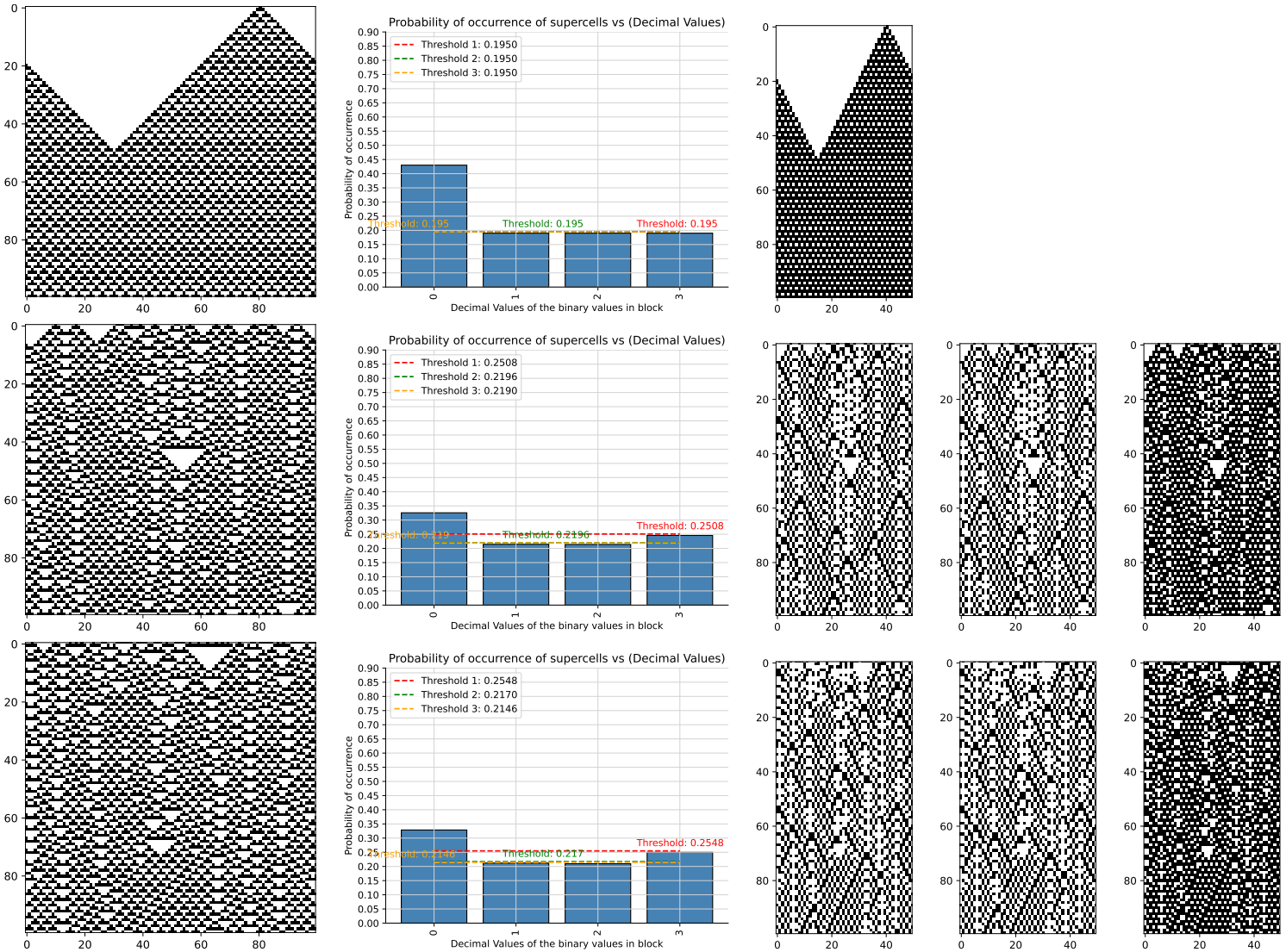

Table 45: FHCG plots for ECA Rule 56.

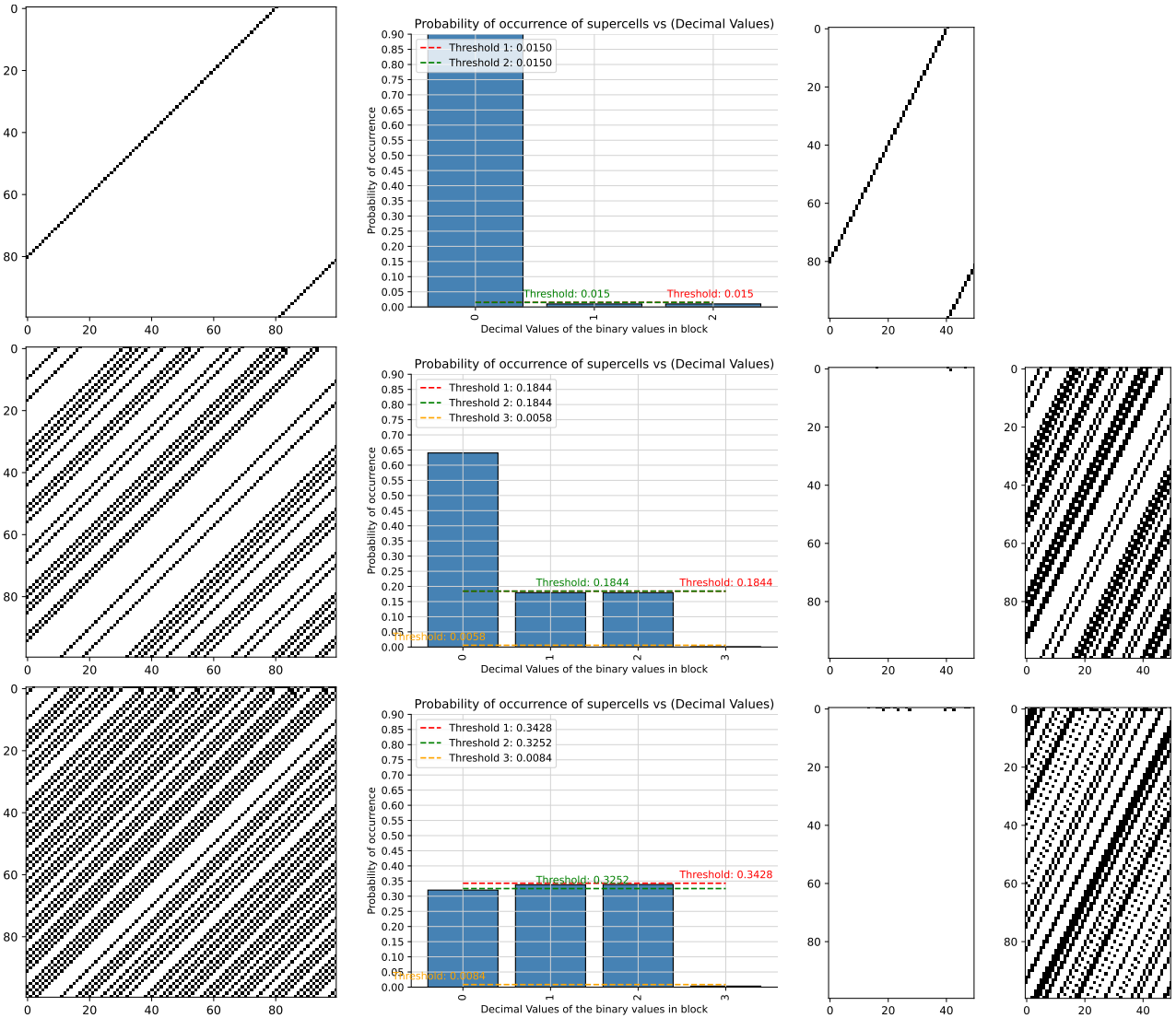

Table 46: FHCG plots for ECA Rule 57.

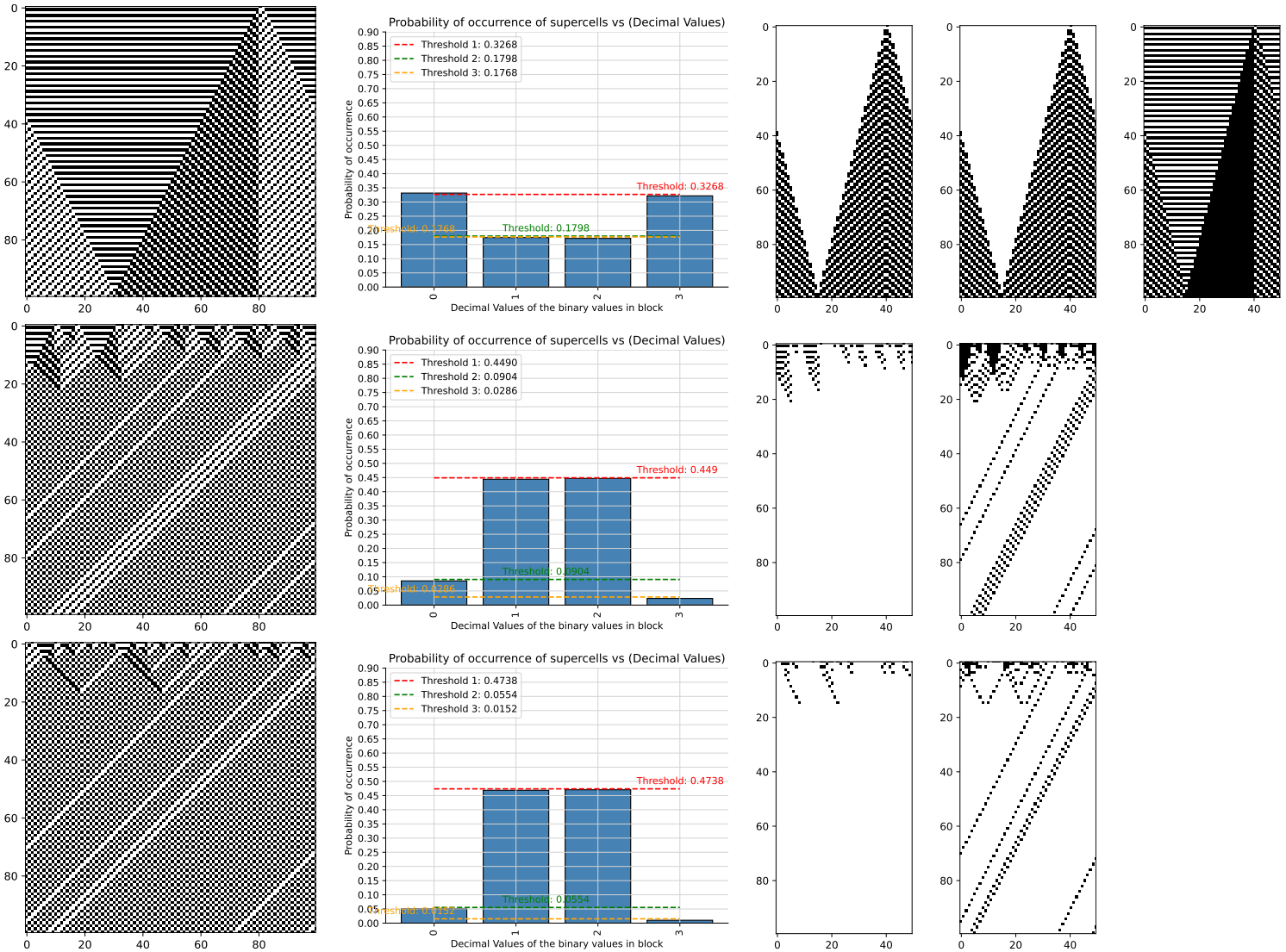

Table 47: FHCG plots for ECA Rule 58.

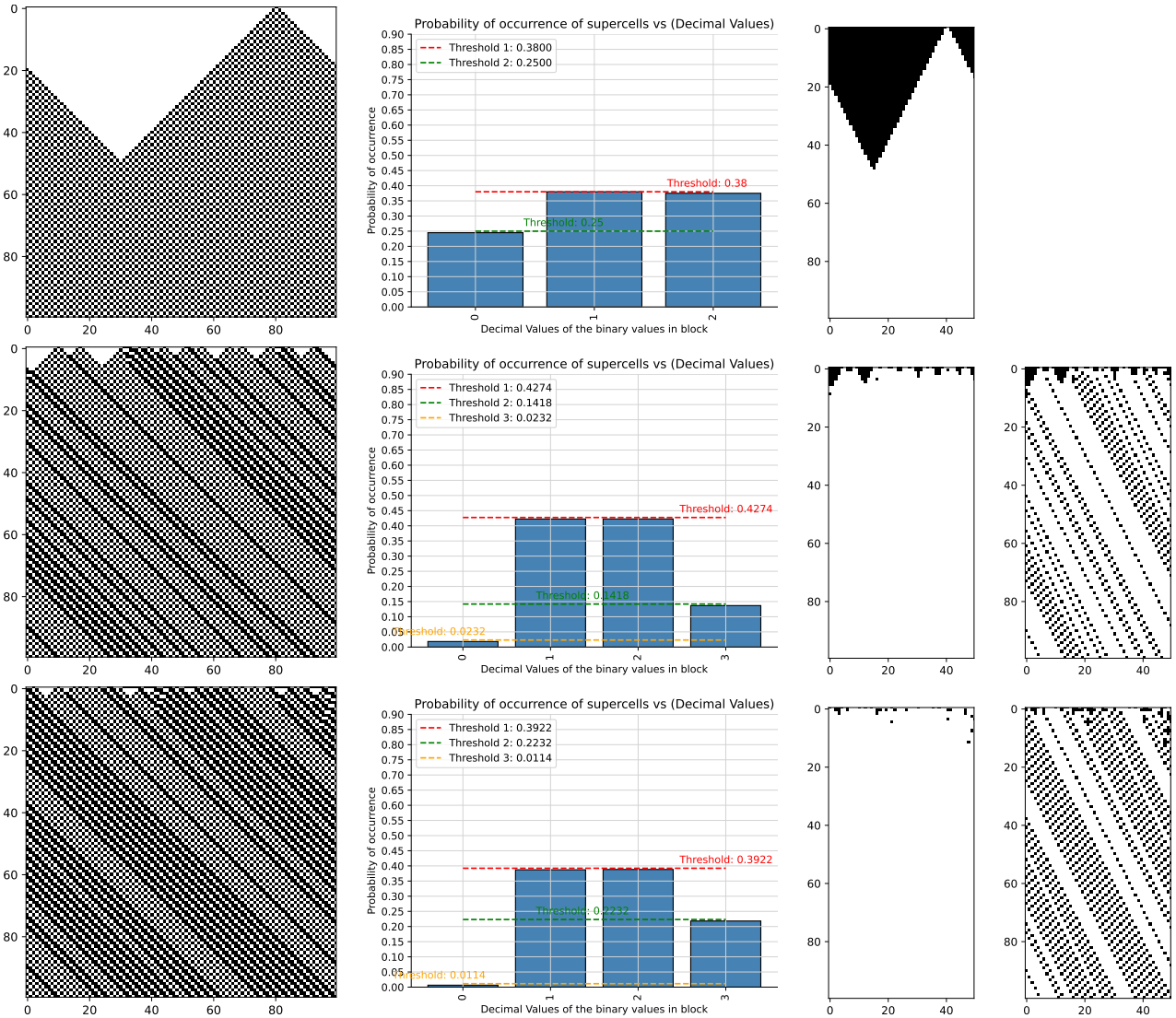

Table 48: FHCG plots for ECA Rule 60.

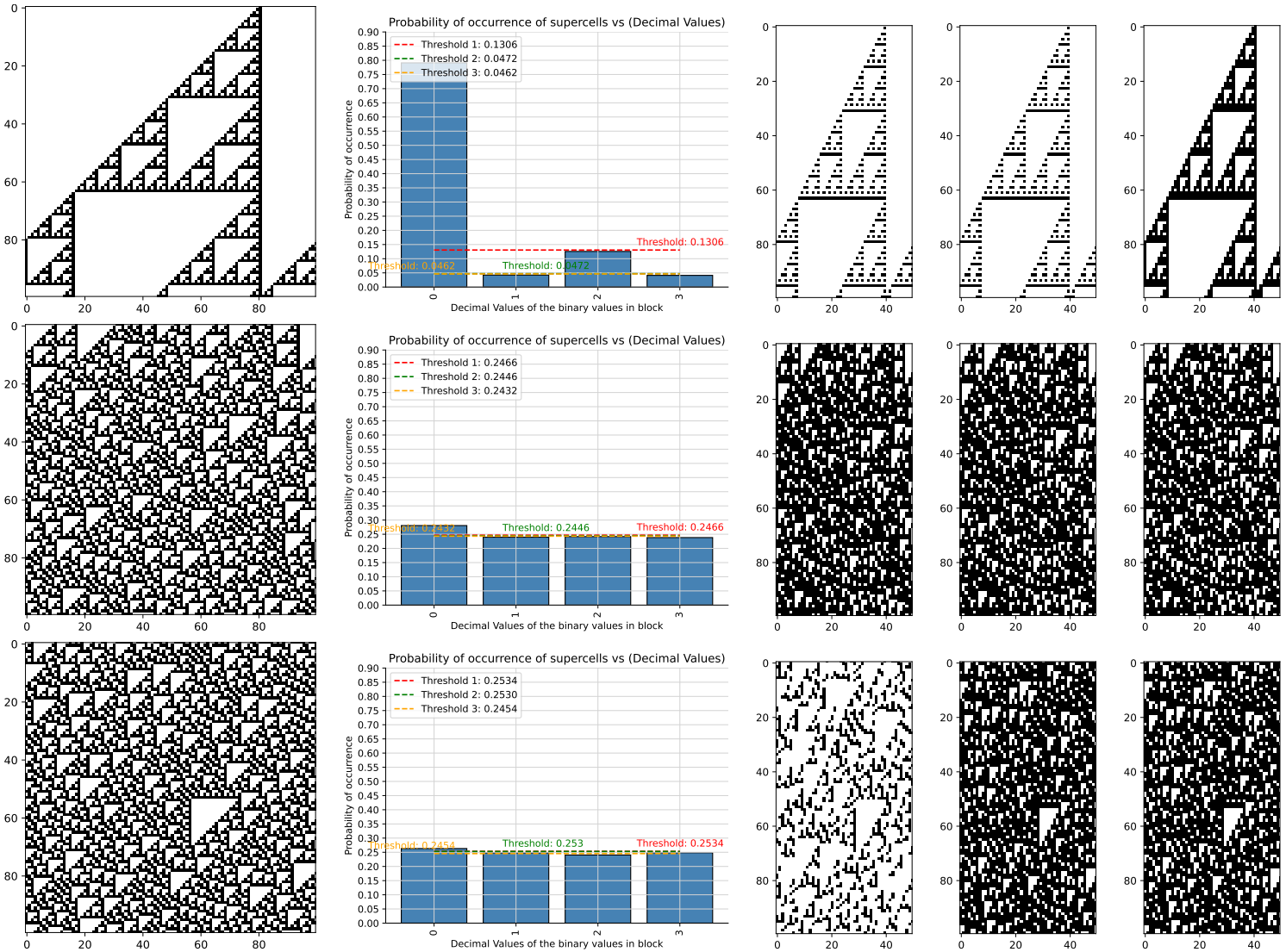

Table 49: FHCG plots for ECA Rule 62.

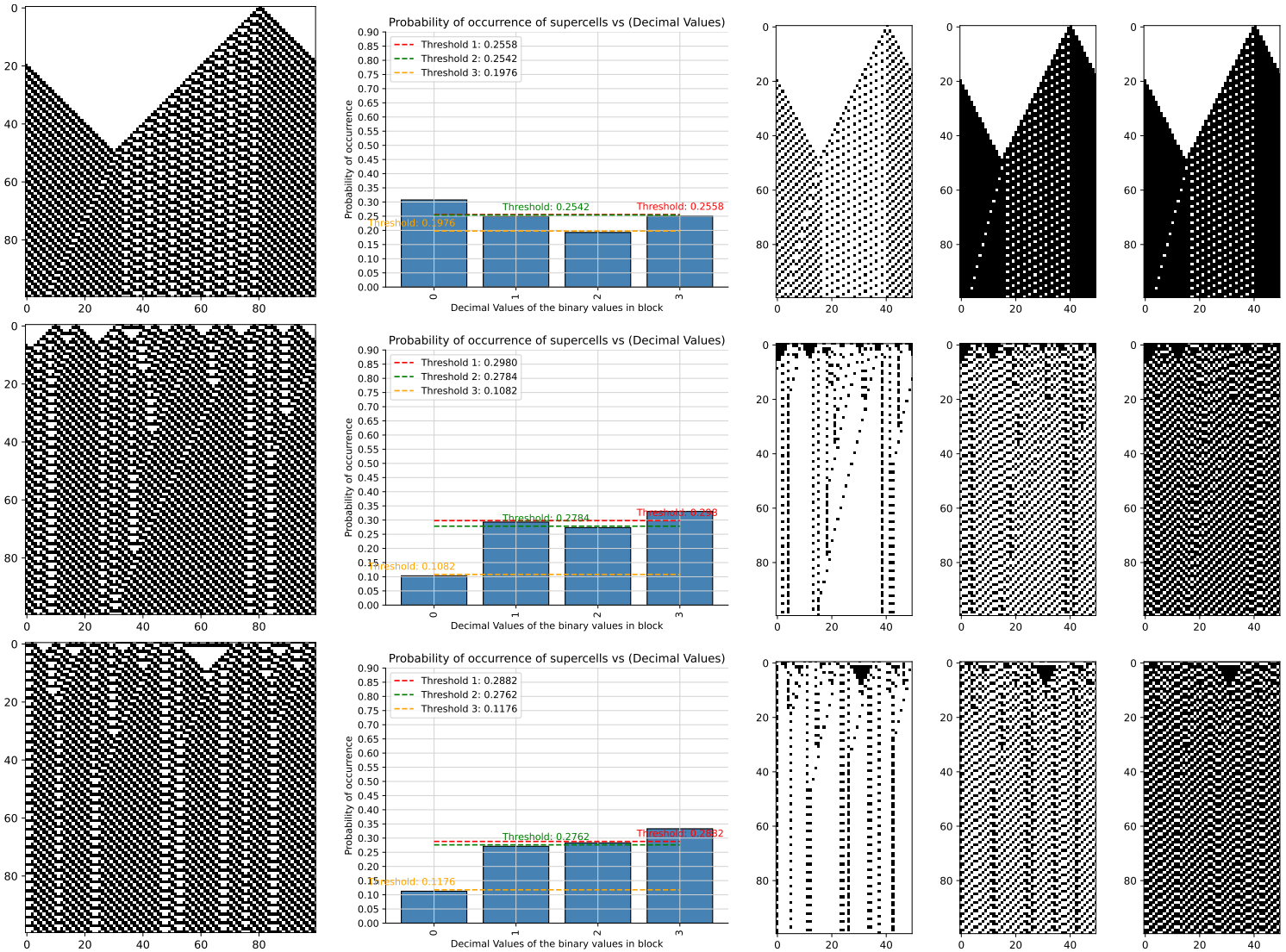

Table 50: FHCG plots for ECA Rule 72.

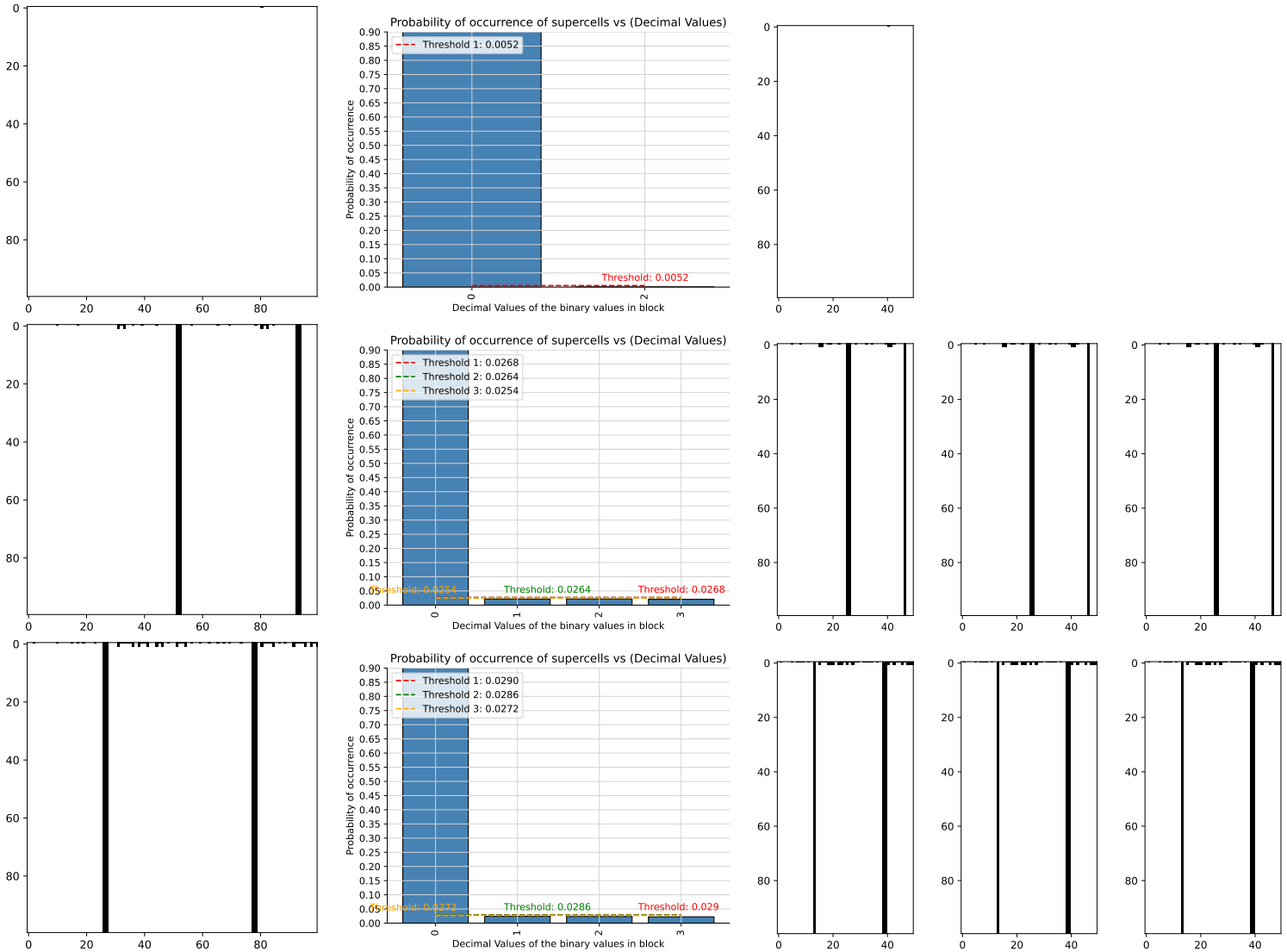

Table 51: FHCG plots for ECA Rule 73.

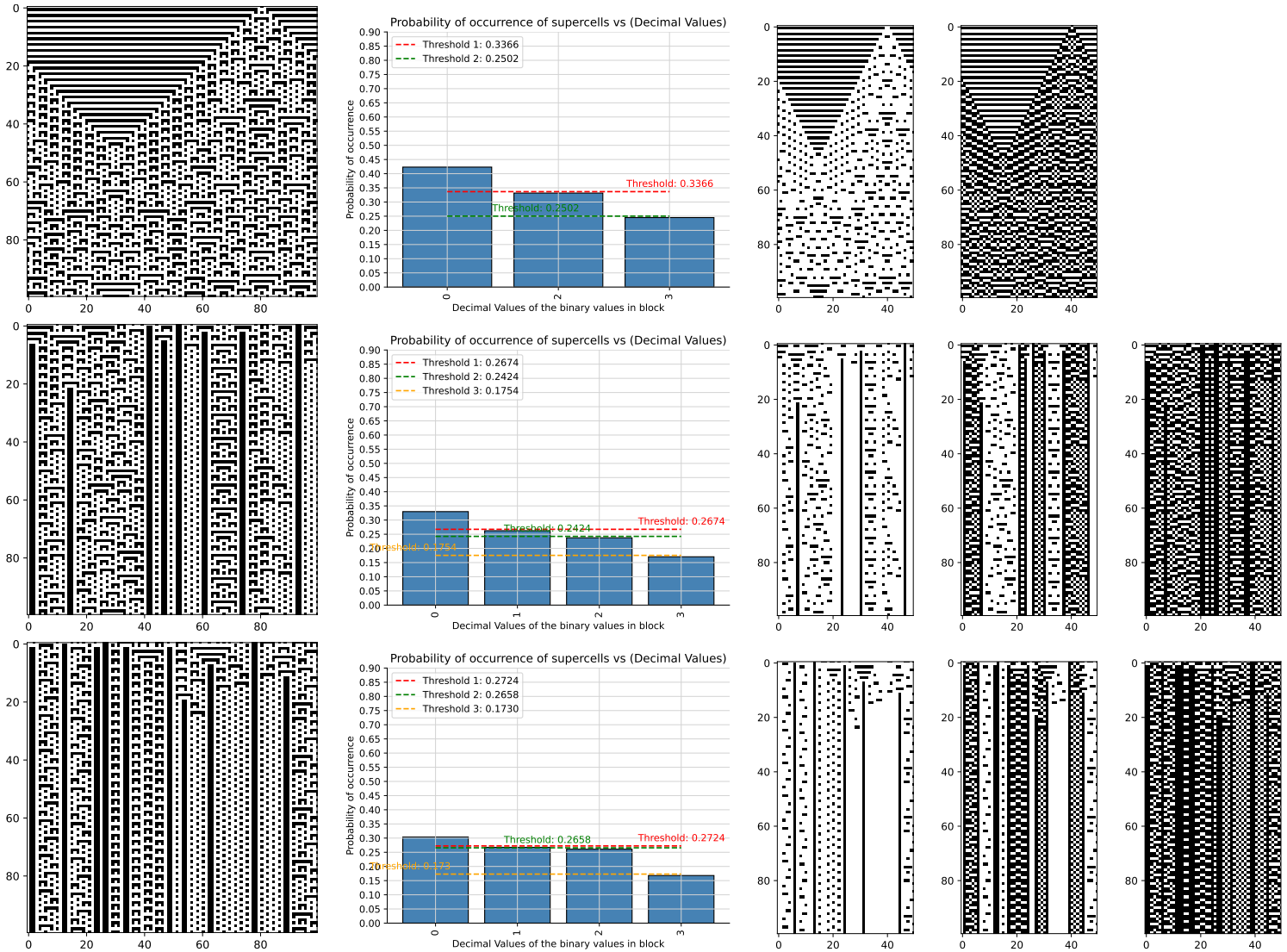

Table 52: FHCG plots for ECA Rule 74.

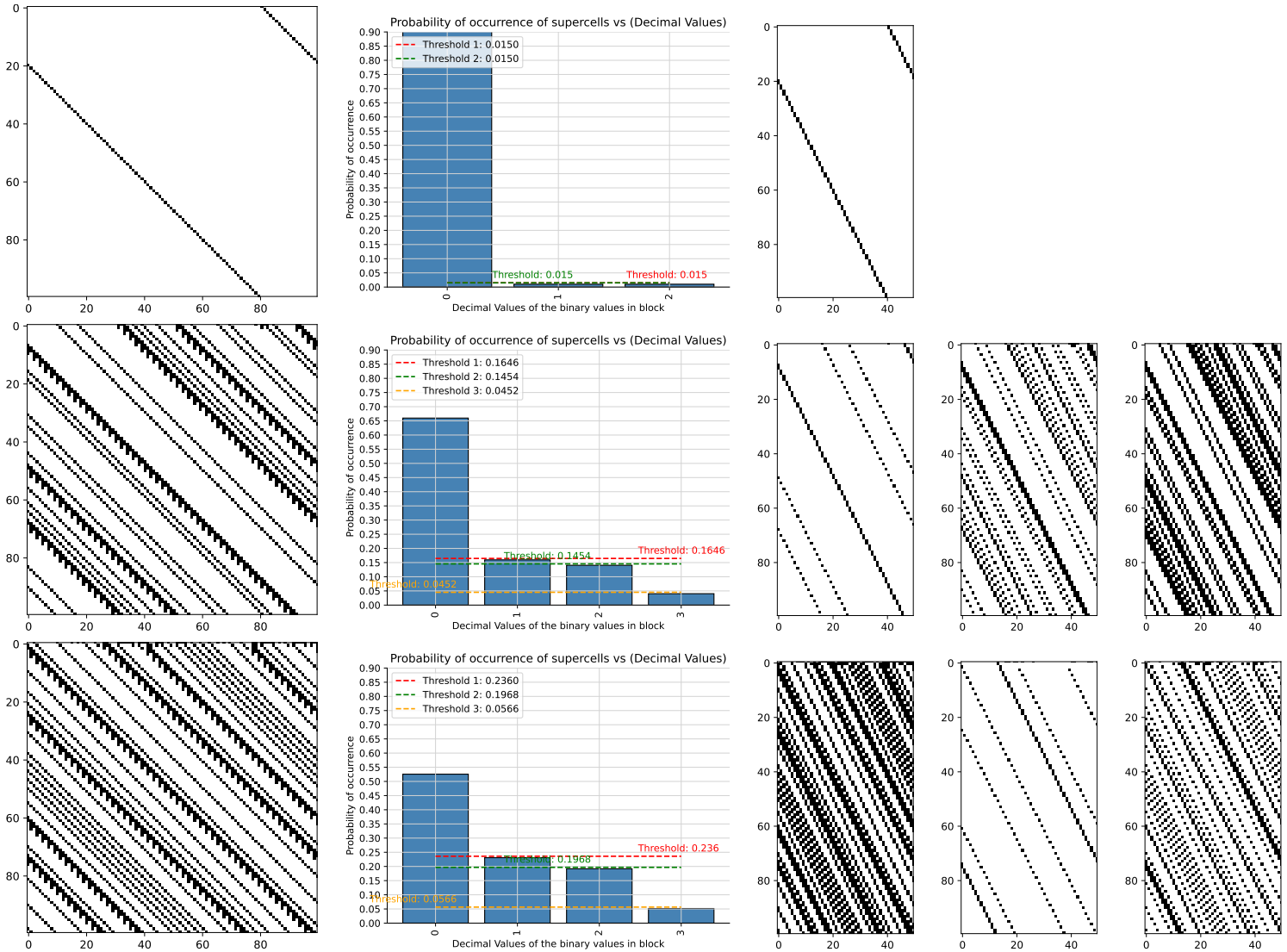

Table 53: FHCG plots for ECA Rule 76.

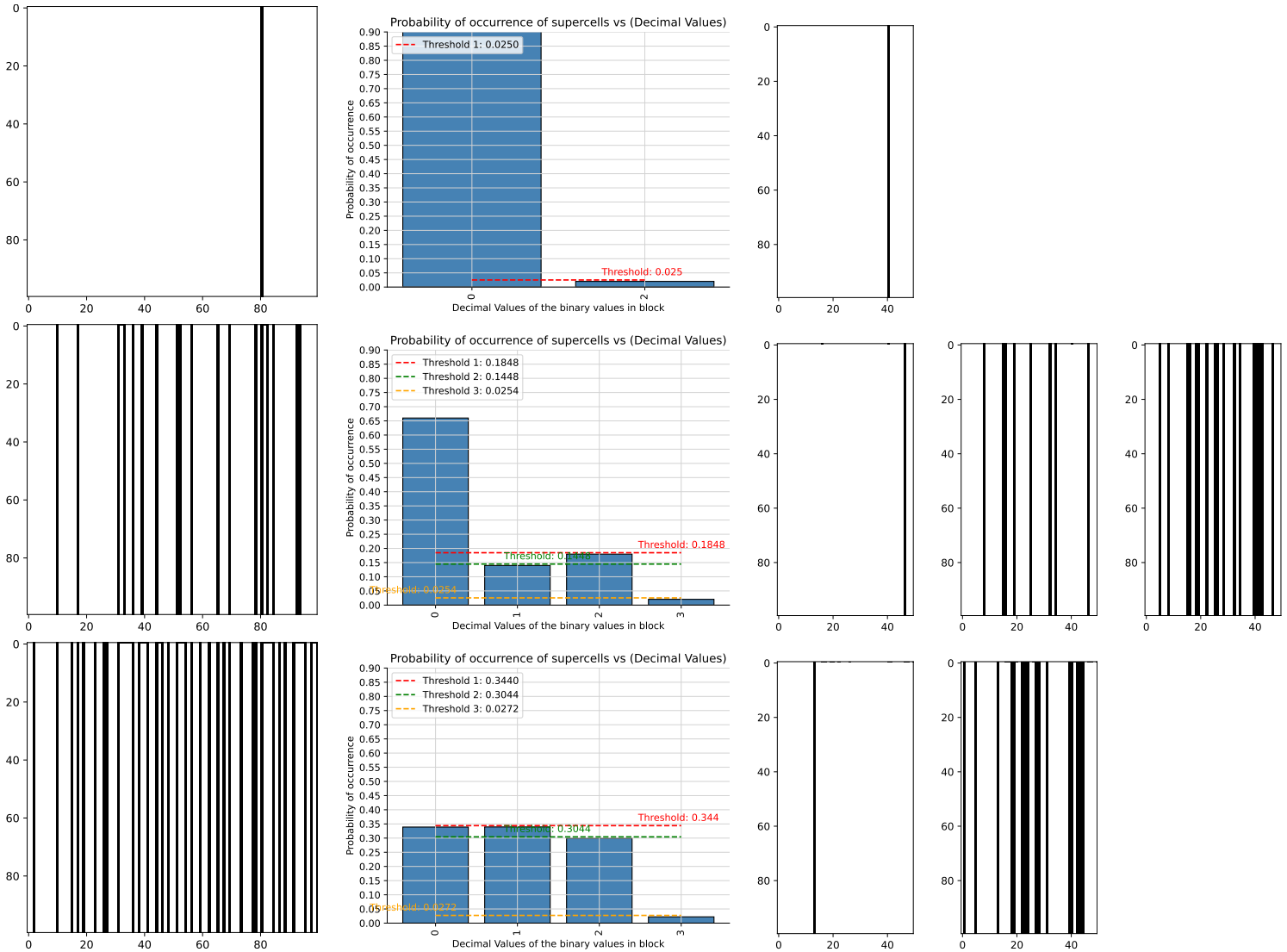

Table 54: FHCG plots for ECA Rule 77.

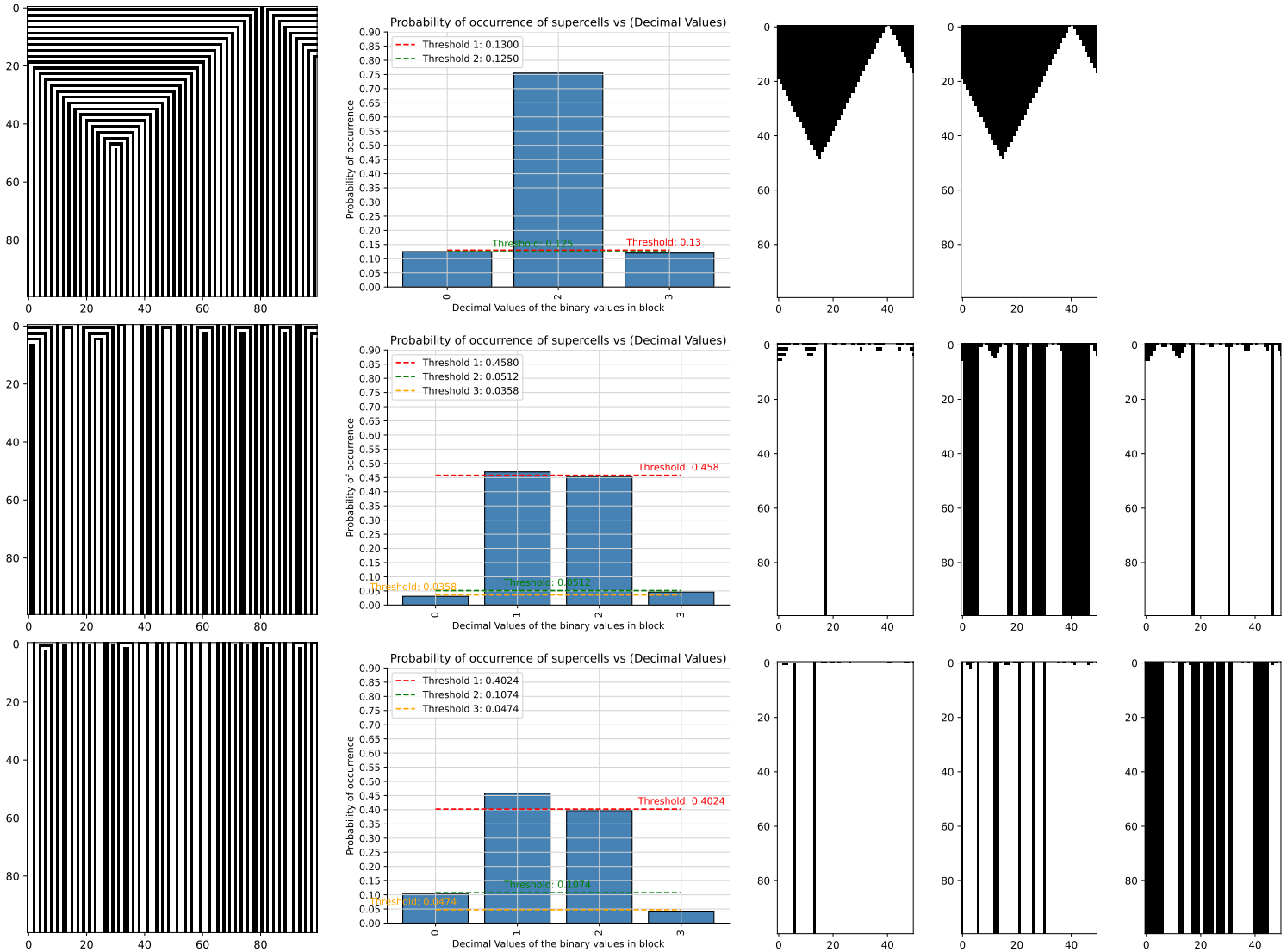

Table 55: FHCG plots for ECA Rule 78.

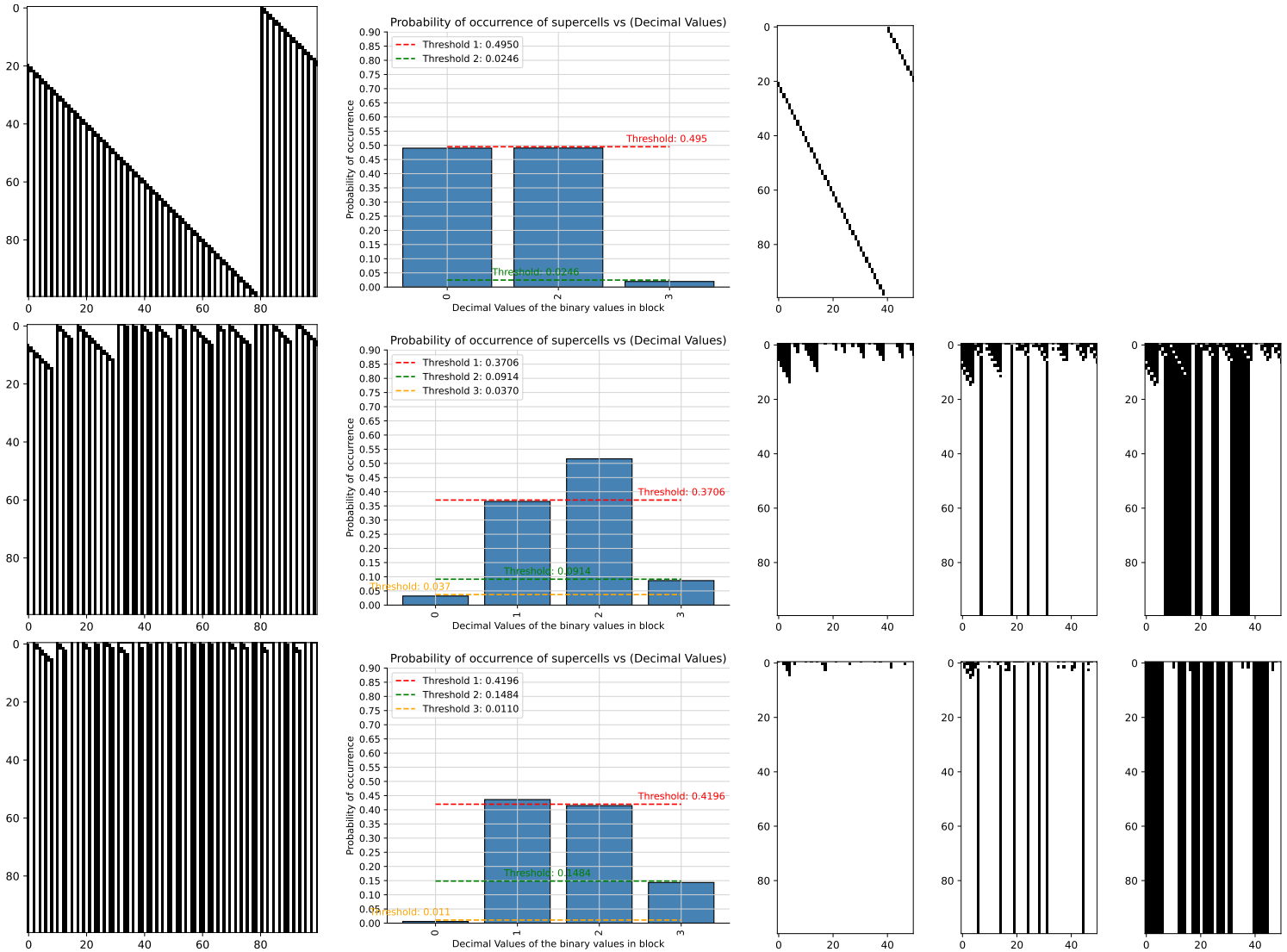

Table 56: FHCG plots for ECA Rule 90.

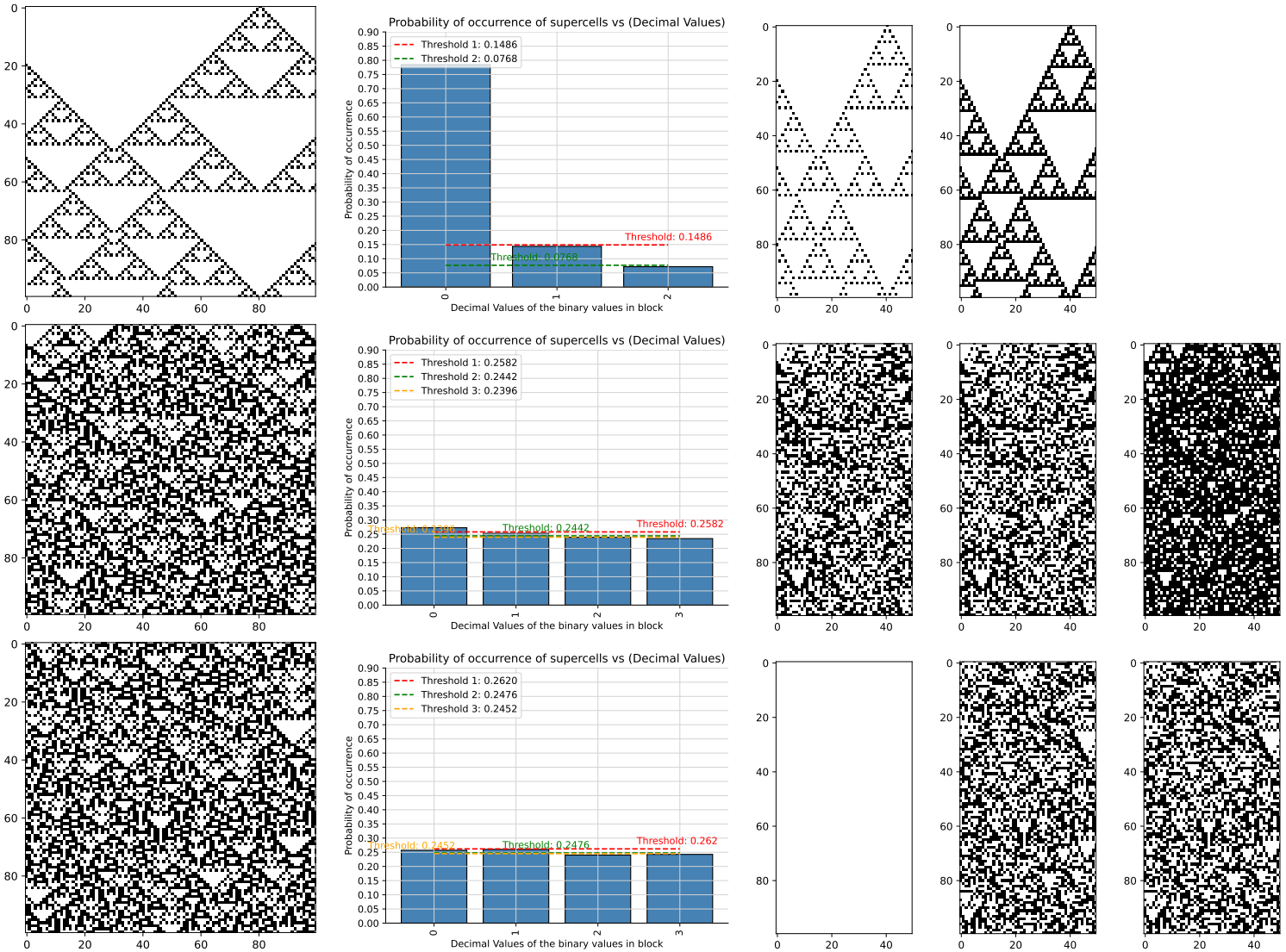

Table 57: FHCG plots for ECA Rule 94.

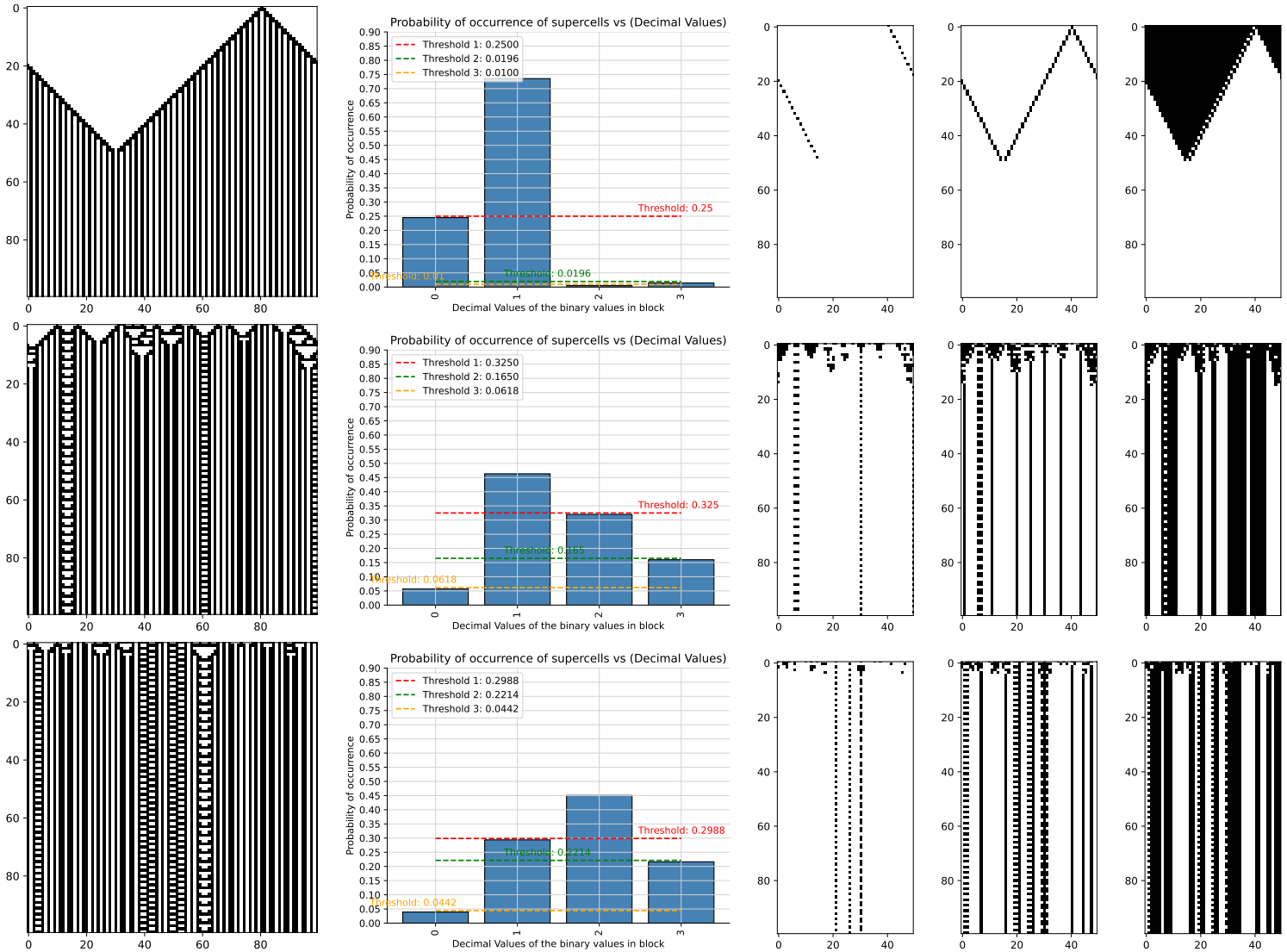

Table 58: FHCG plots for ECA Rule 104.

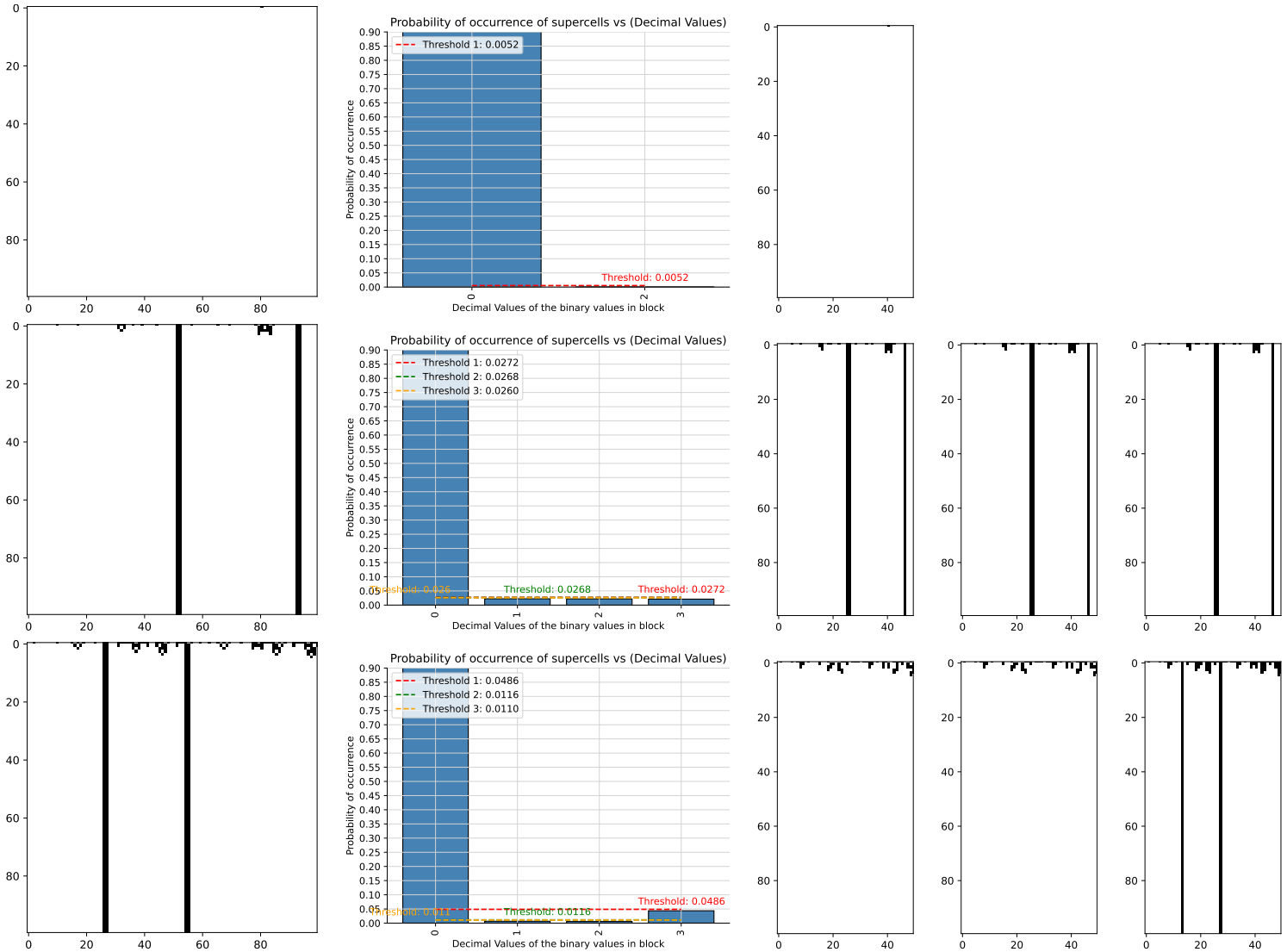

Table 59: FHCG plots for ECA Rule 105.

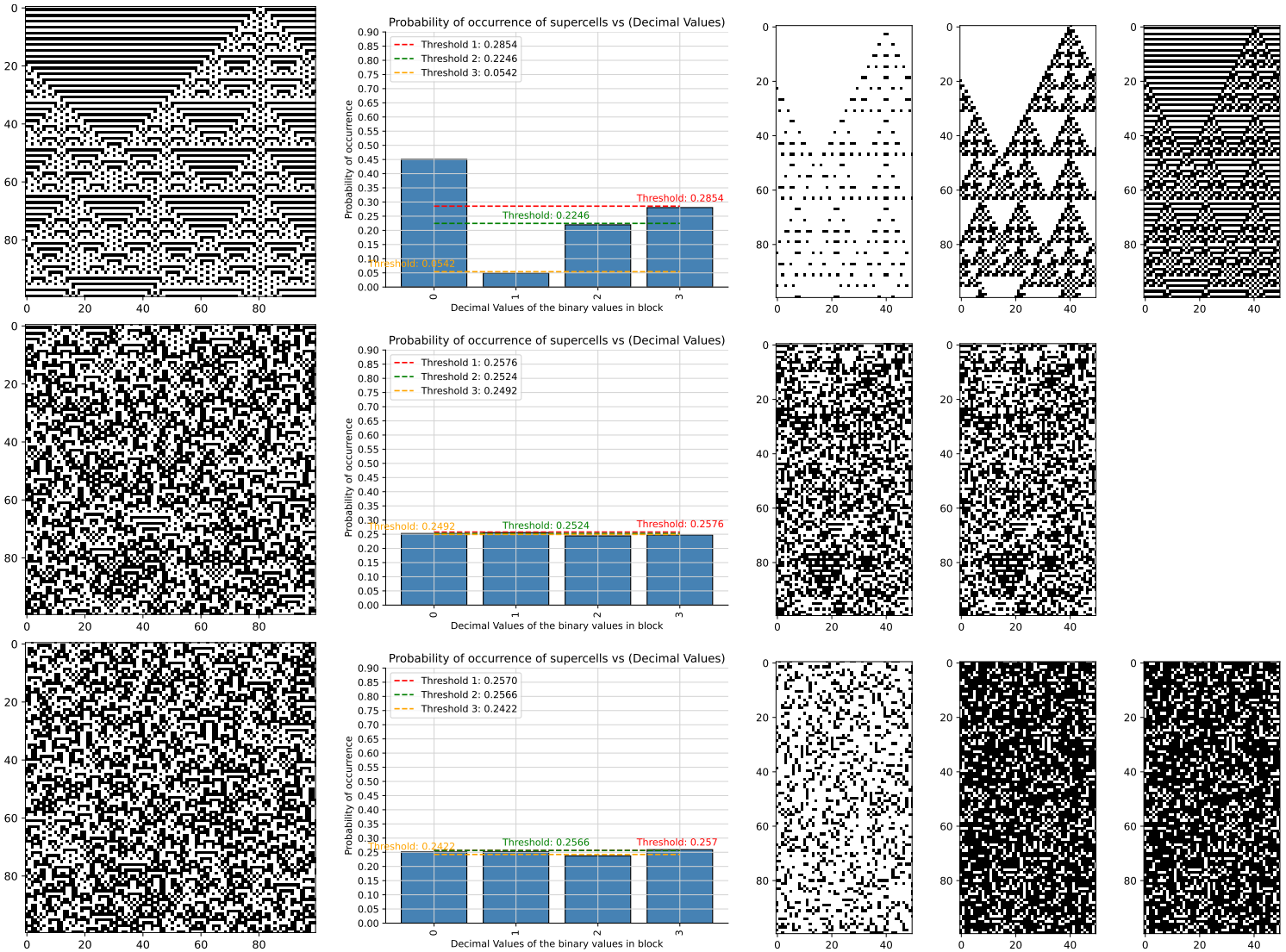

Table 60: FHCG plots for ECA Rule 106.

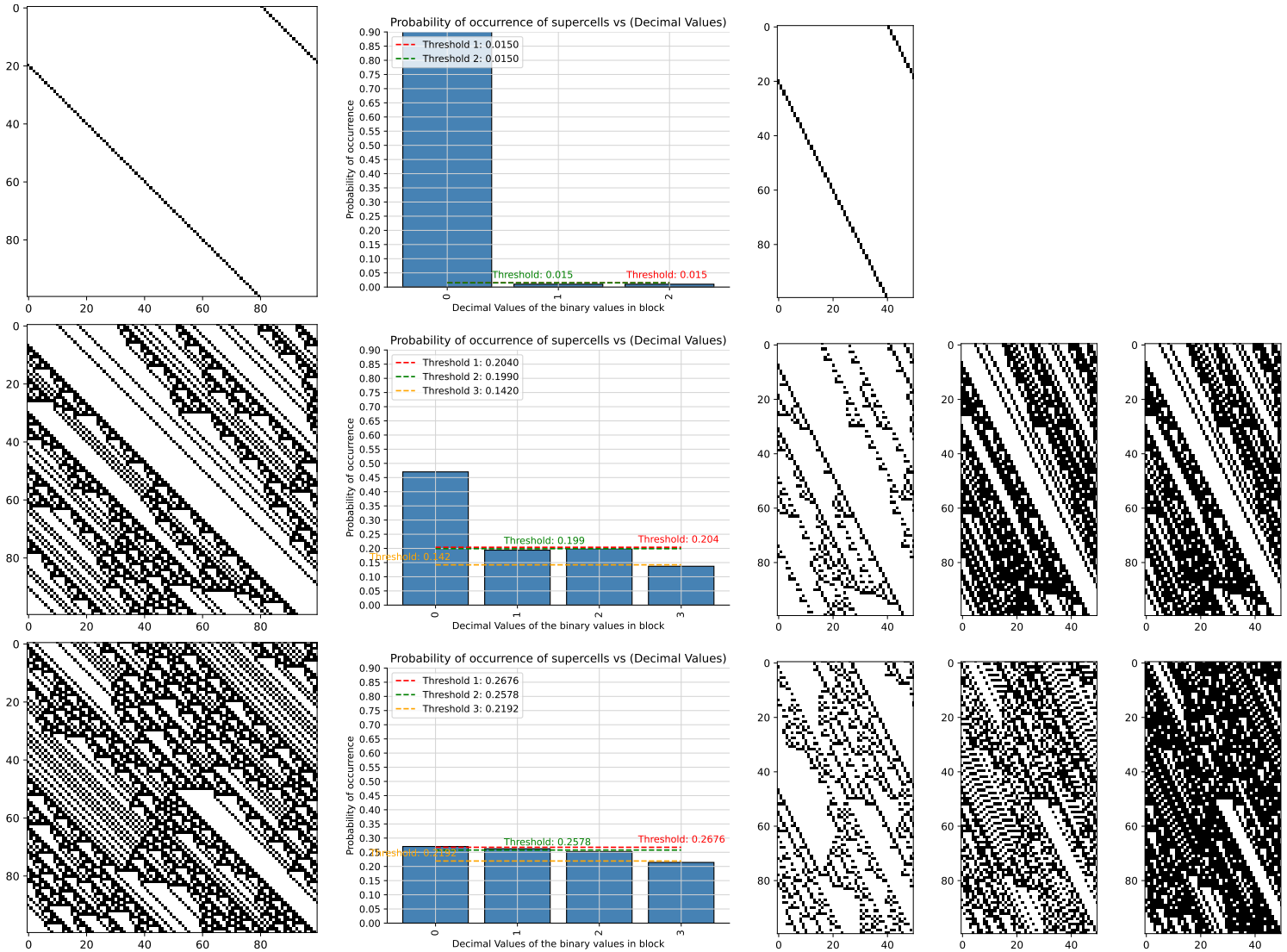

Table 61: FHCG plots for ECA Rule 108.

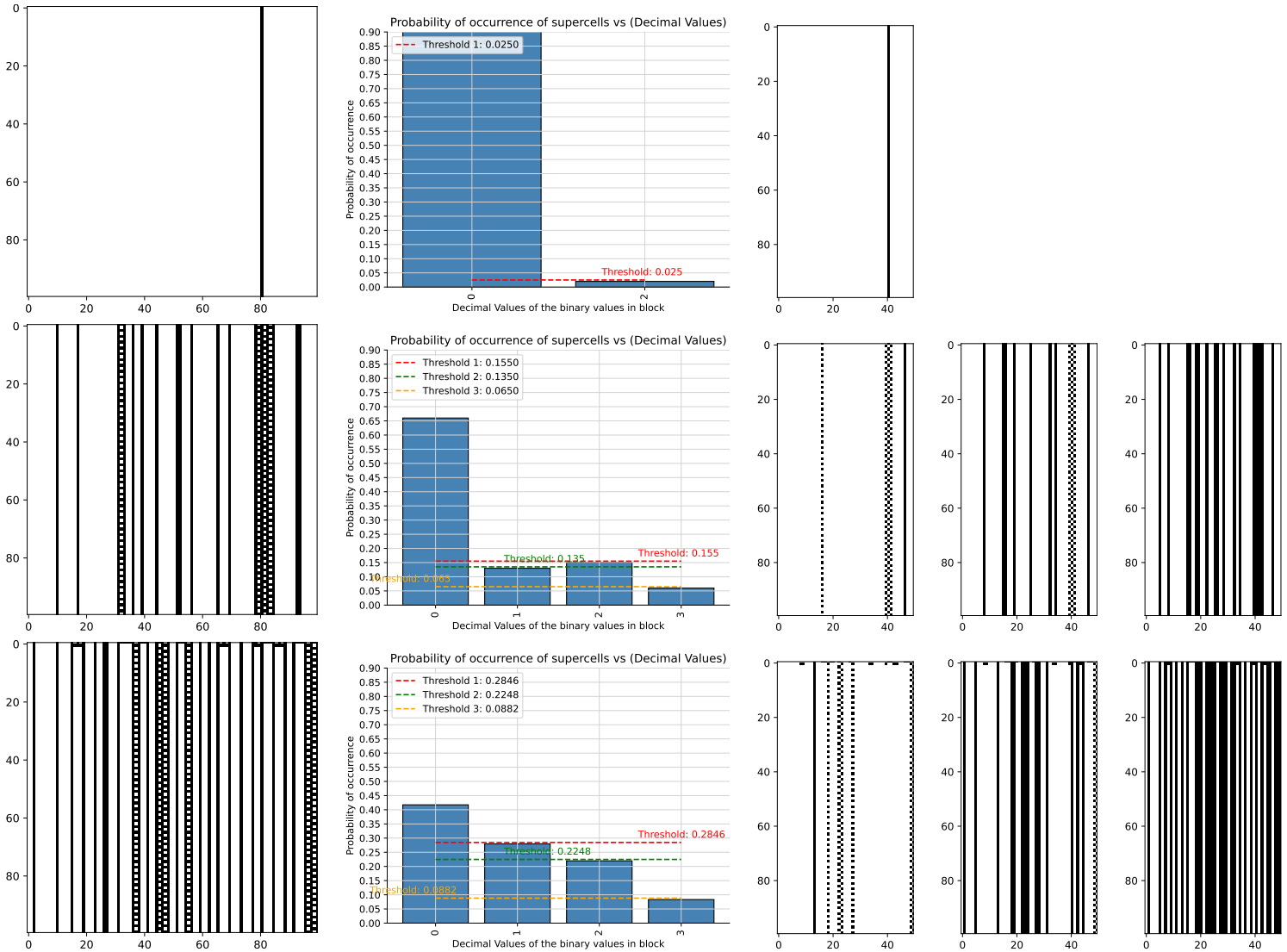

Table 62: FHCG plots for ECA Rule 110.

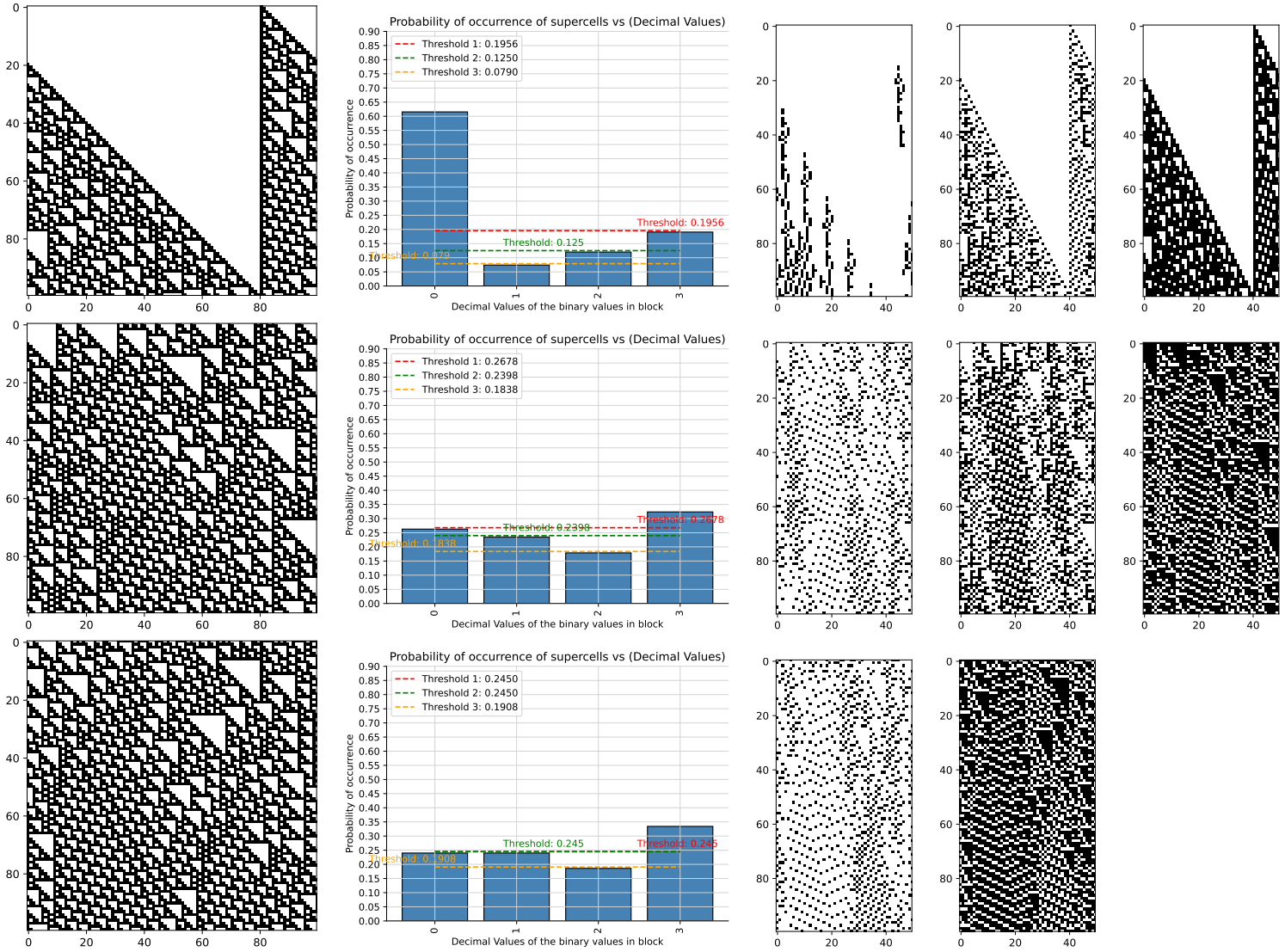

Table 63: FHCG plots for ECA Rule 122.

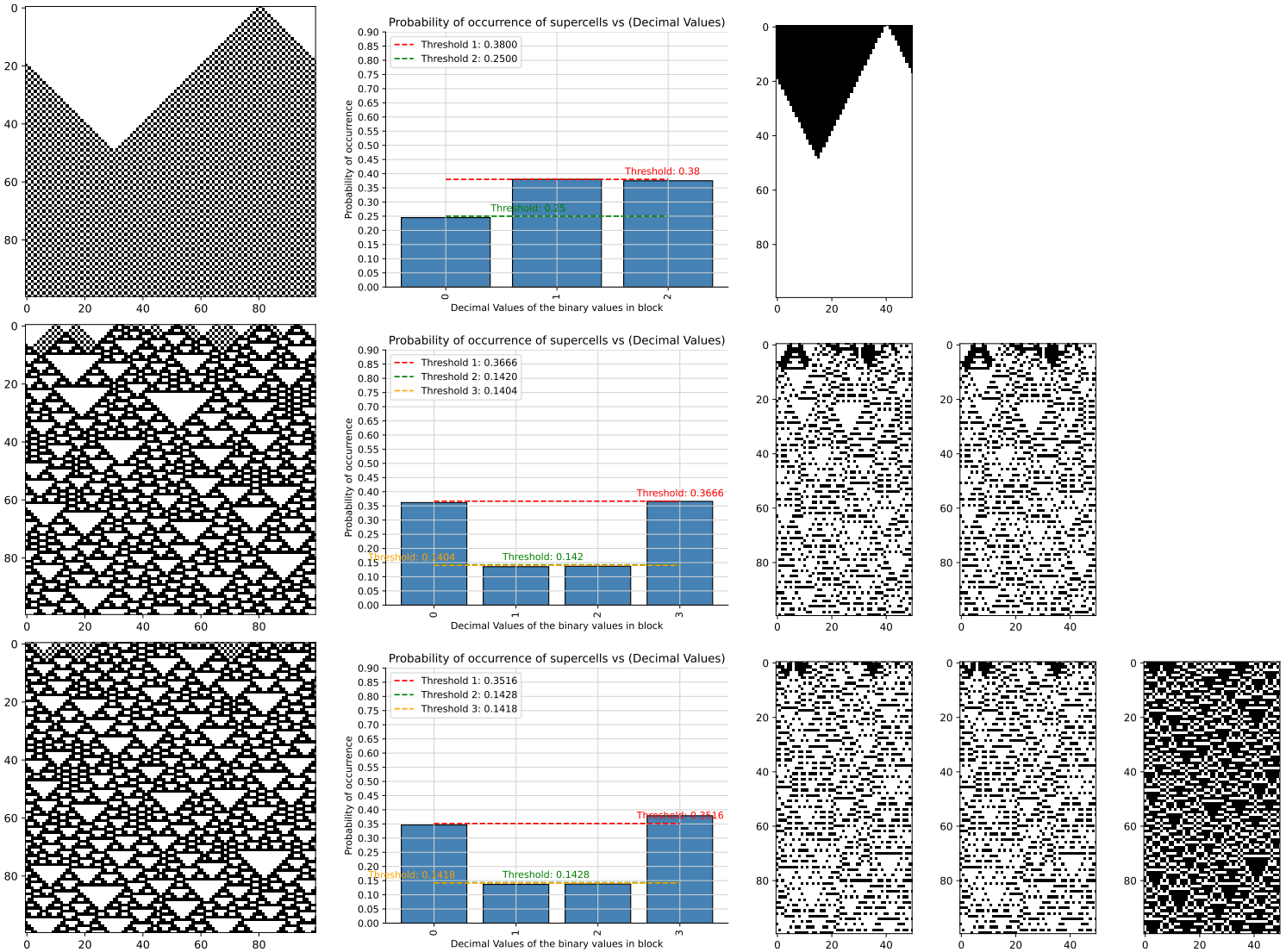

Table 64: FHCG plots for ECA Rule 126.

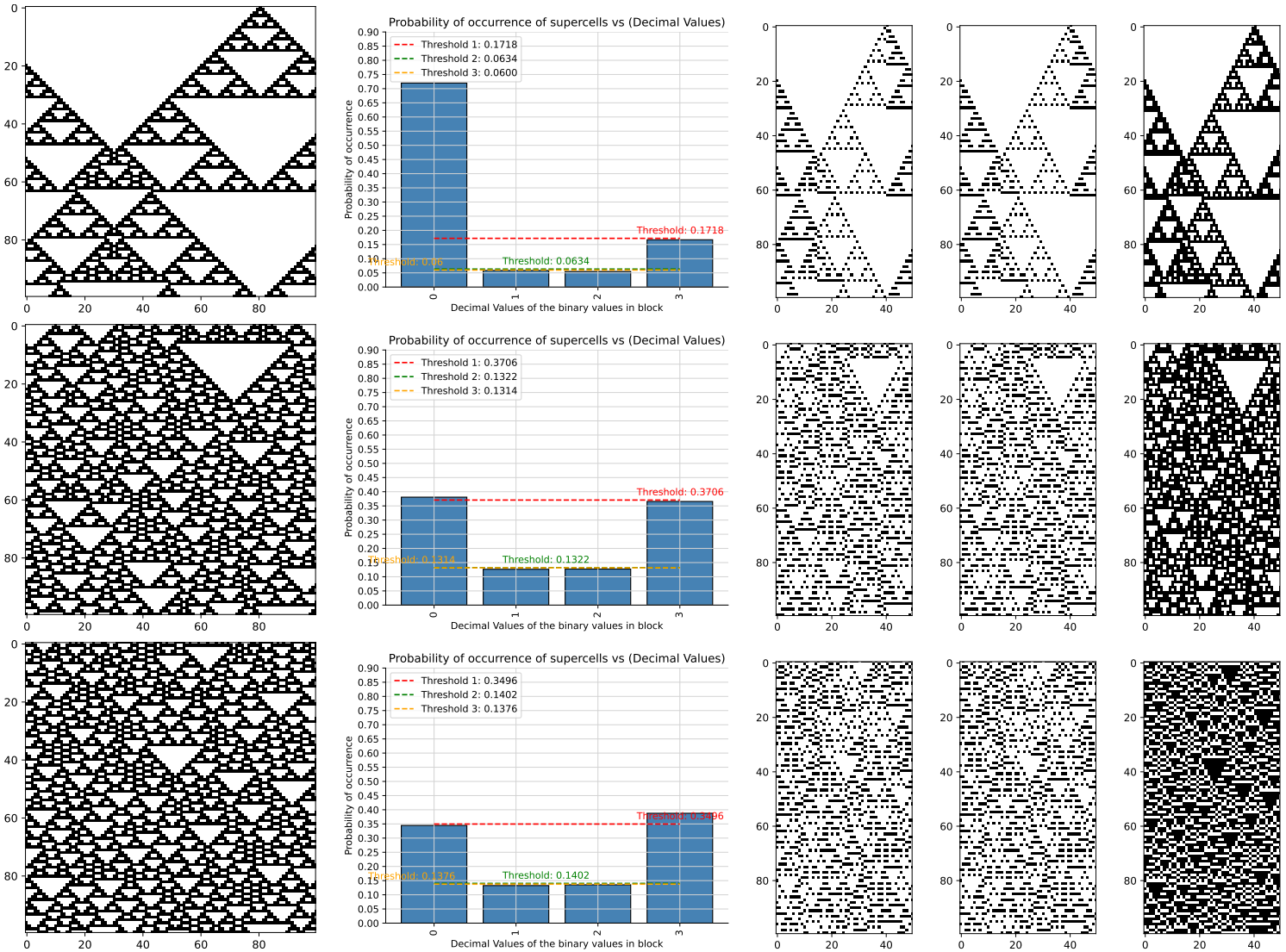

Table 65: FHCG plots for ECA Rule 128.

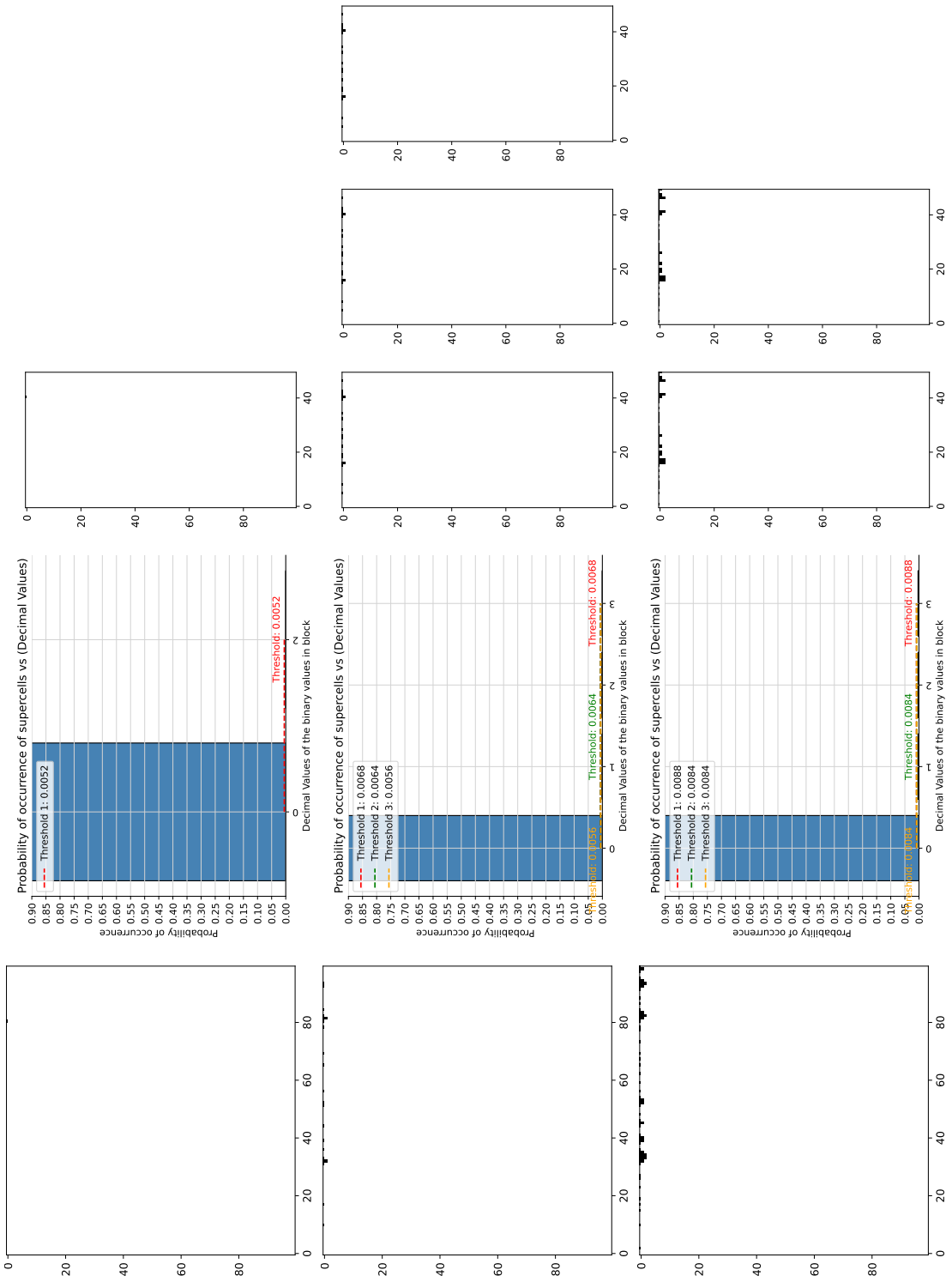

Table 66: FHCG plots for ECA Rule 130.

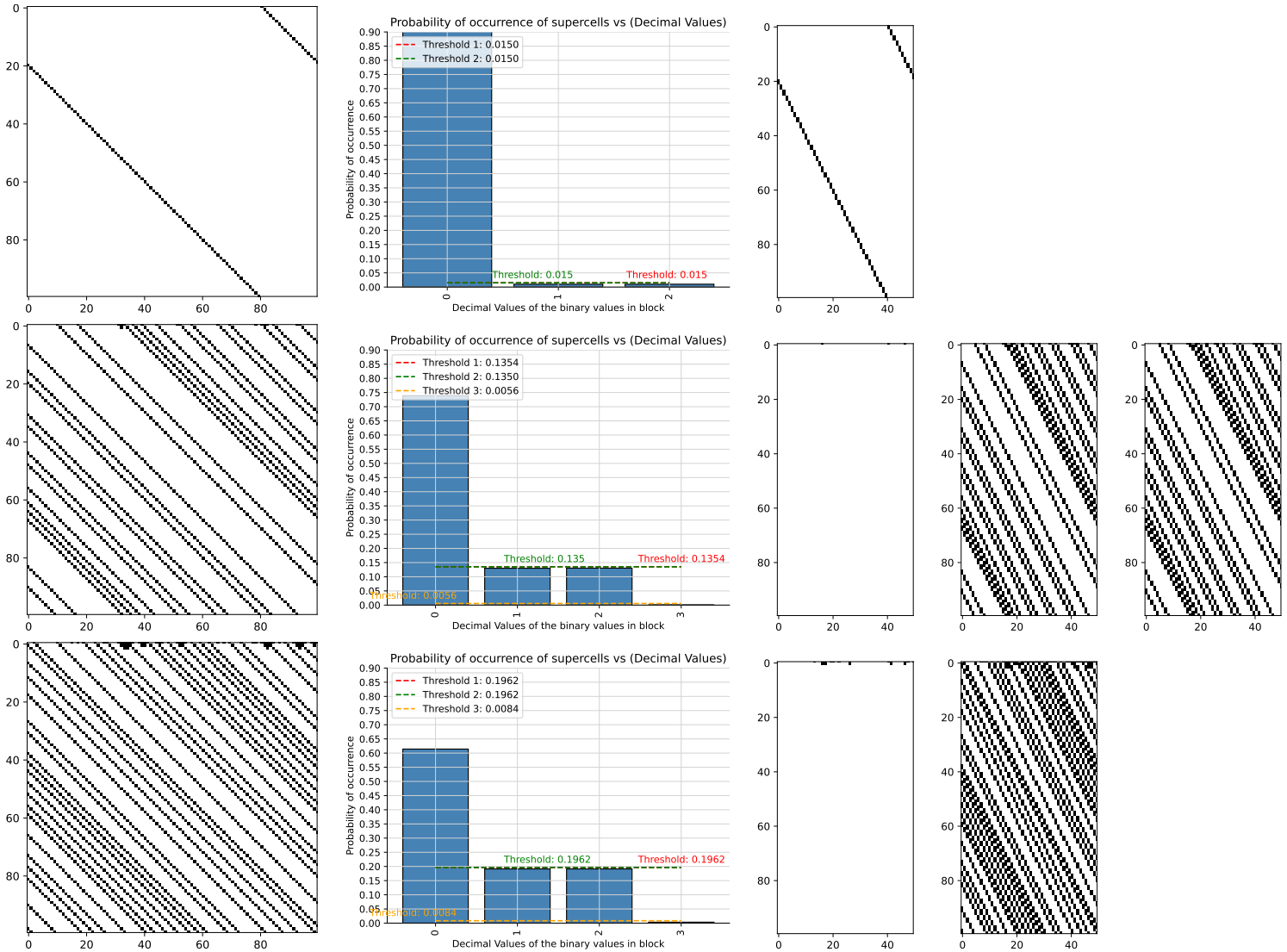

Table 67: FHCG plots for ECA Rule 132.

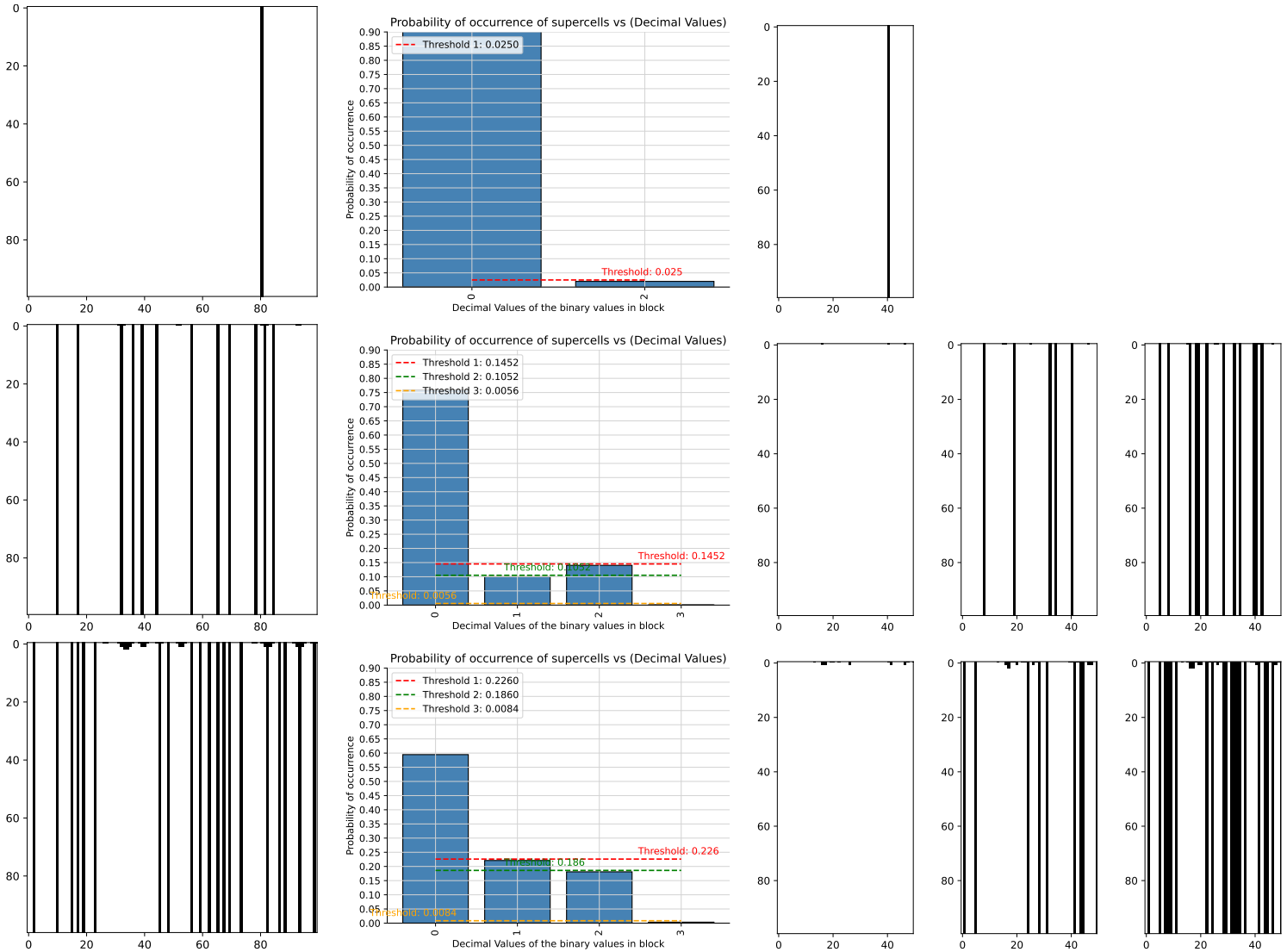

Table 68: FHCG plots for ECA Rule 134.

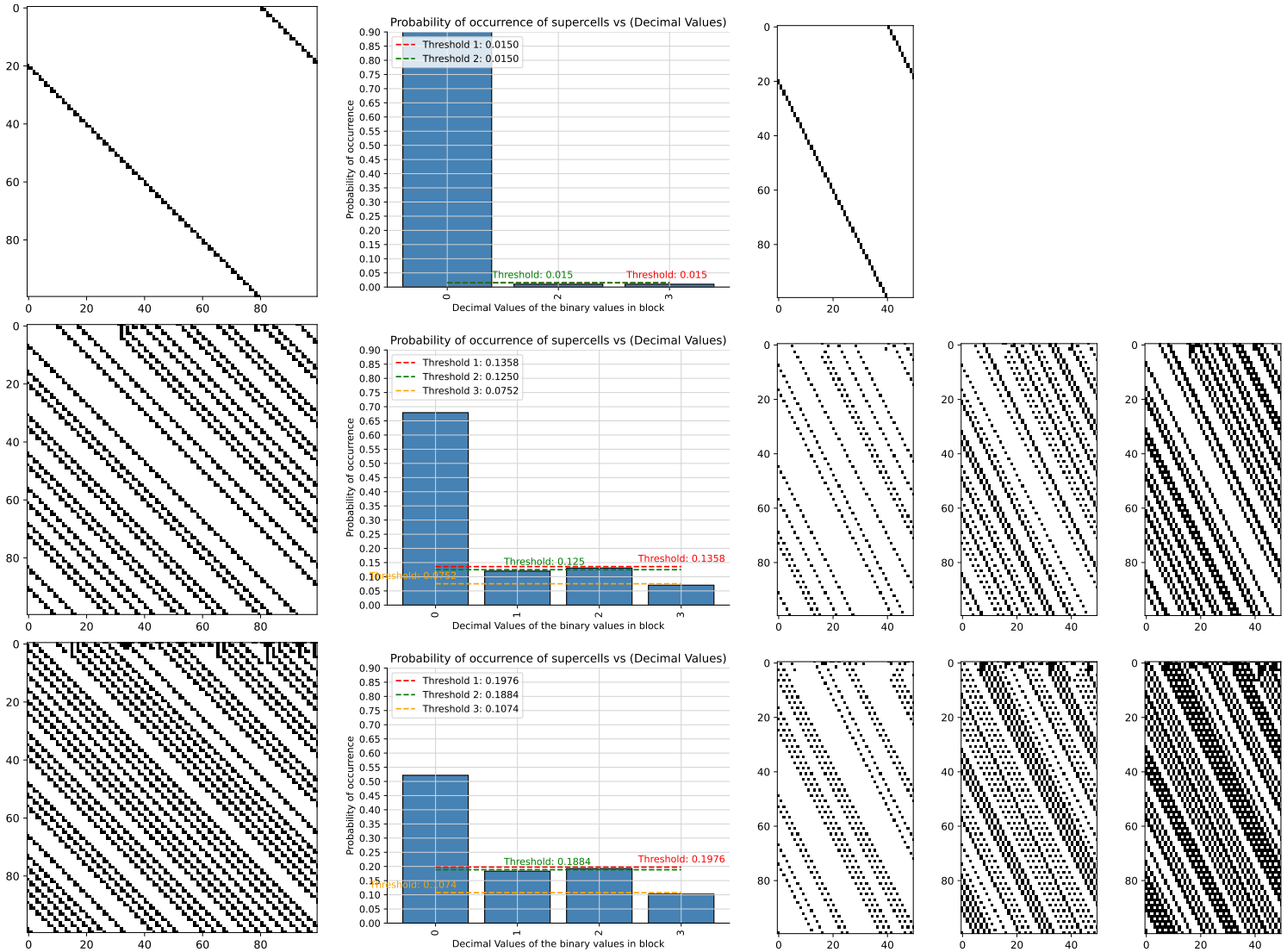

Table 69: FHCG plots for ECA Rule 136.

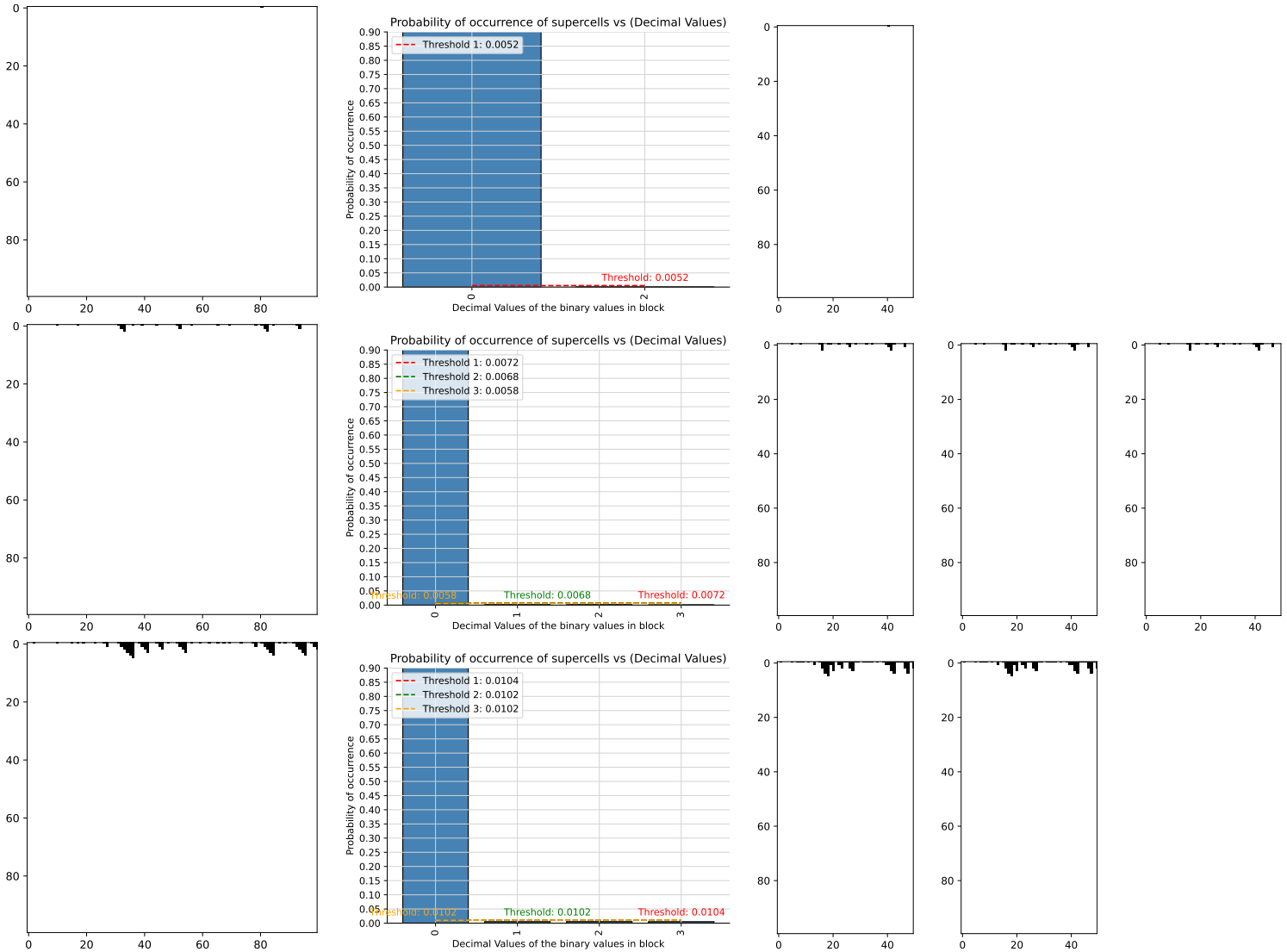

Table 70: FHCG plots for ECA Rule 138.

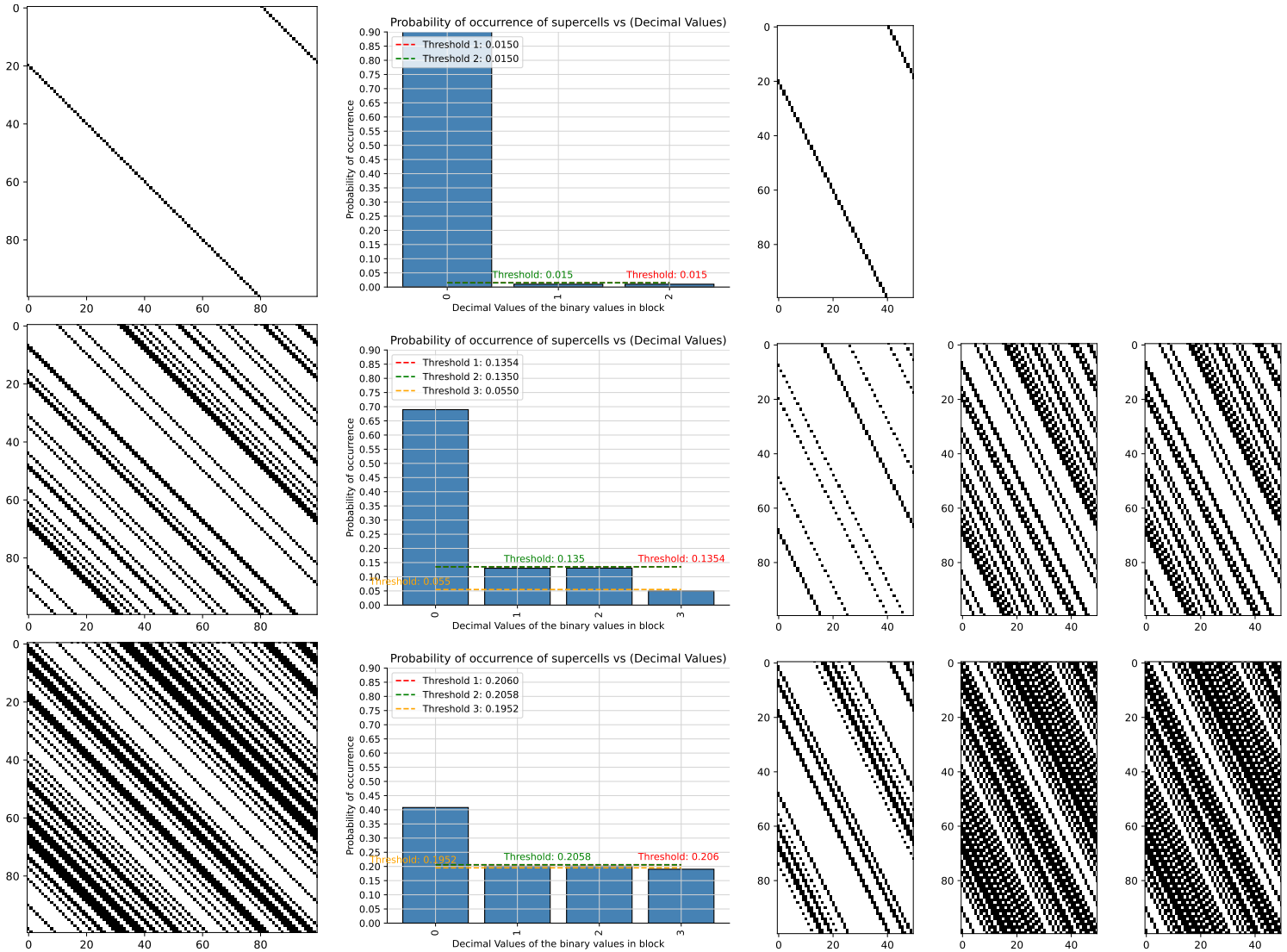

Table 71: FHCG plots for ECA Rule 140.

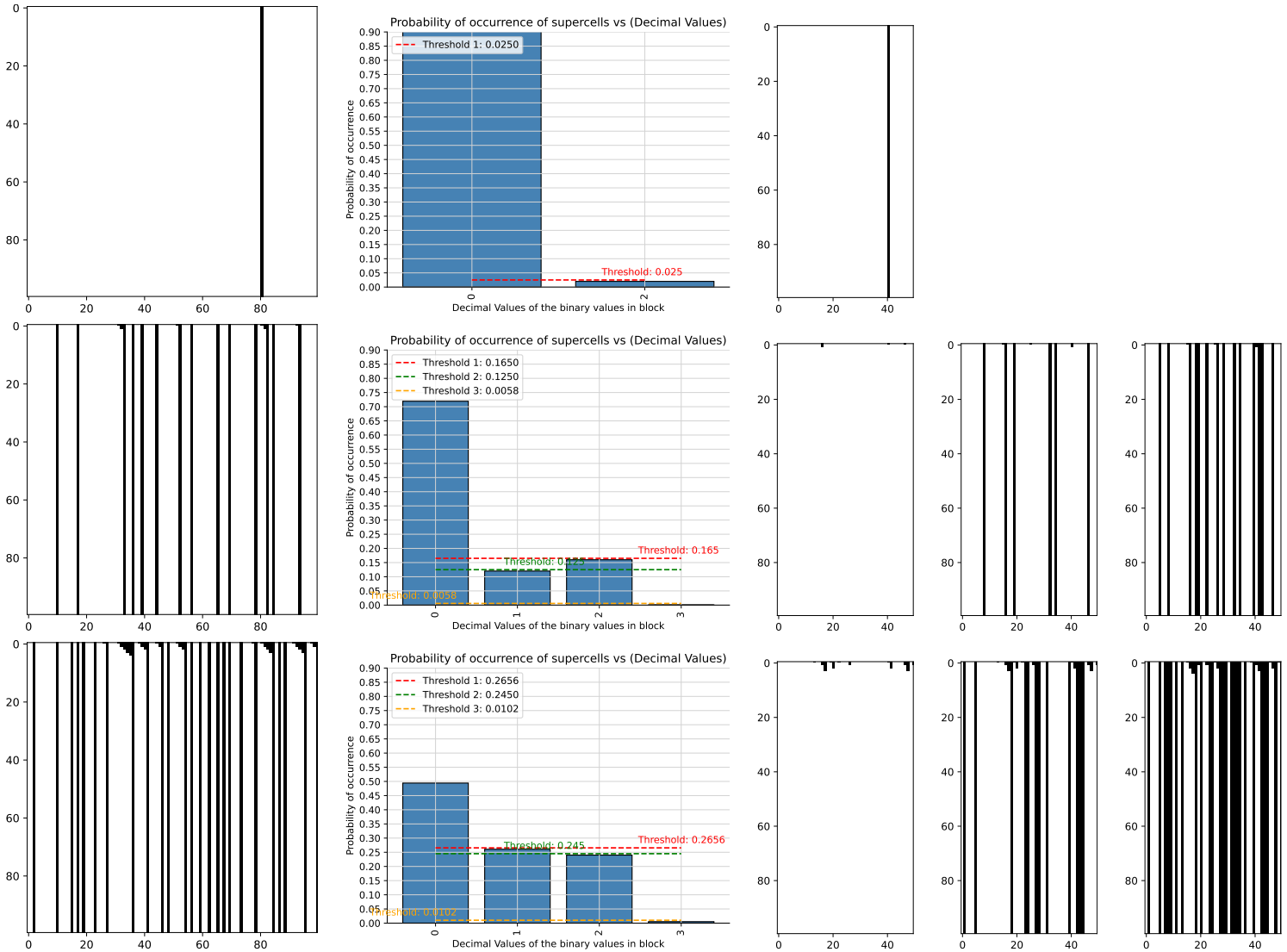

Table 72: FHCG plots for ECA Rule 142.

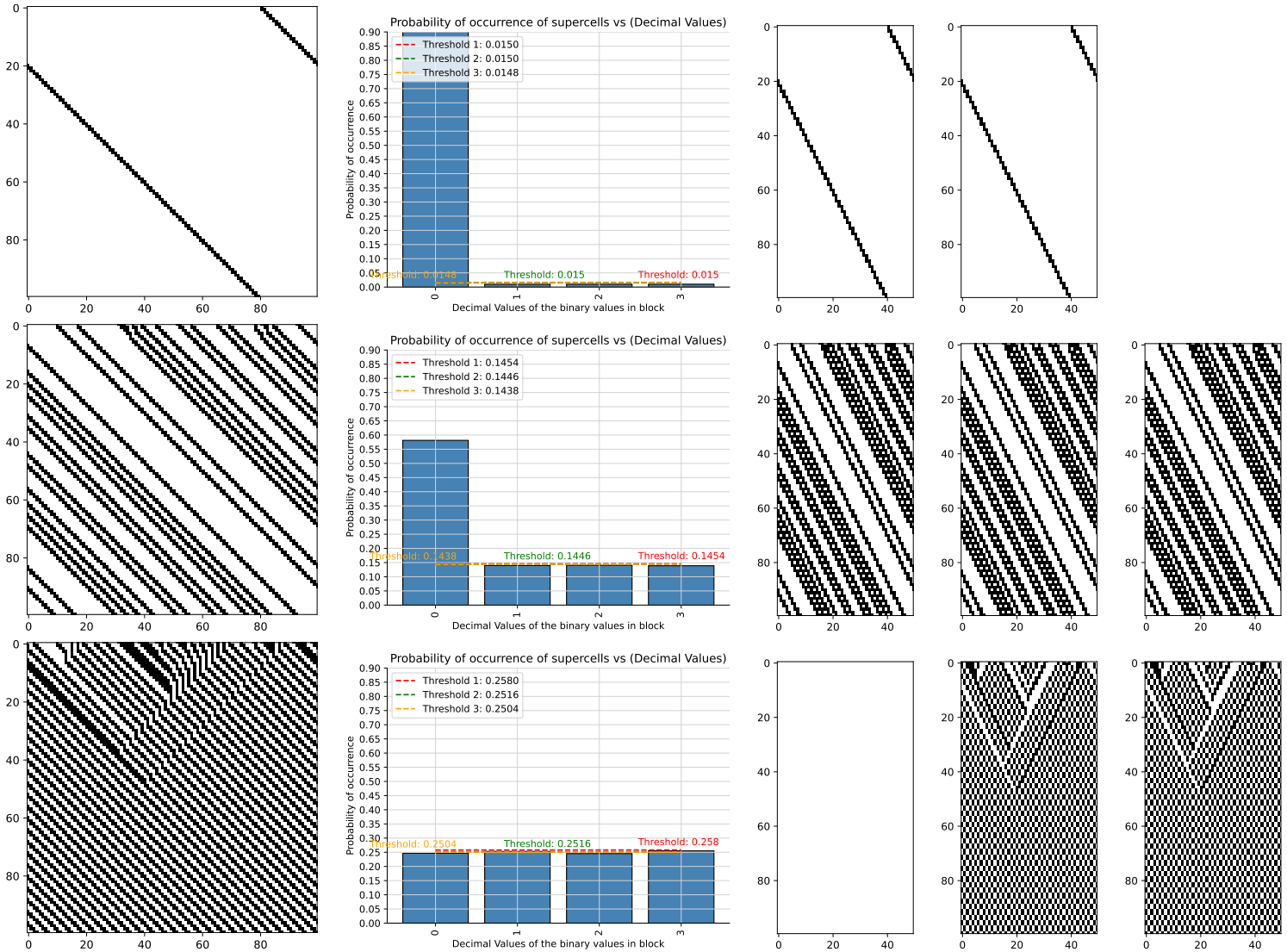

Table 73: FHCG plots for ECA Rule 146.

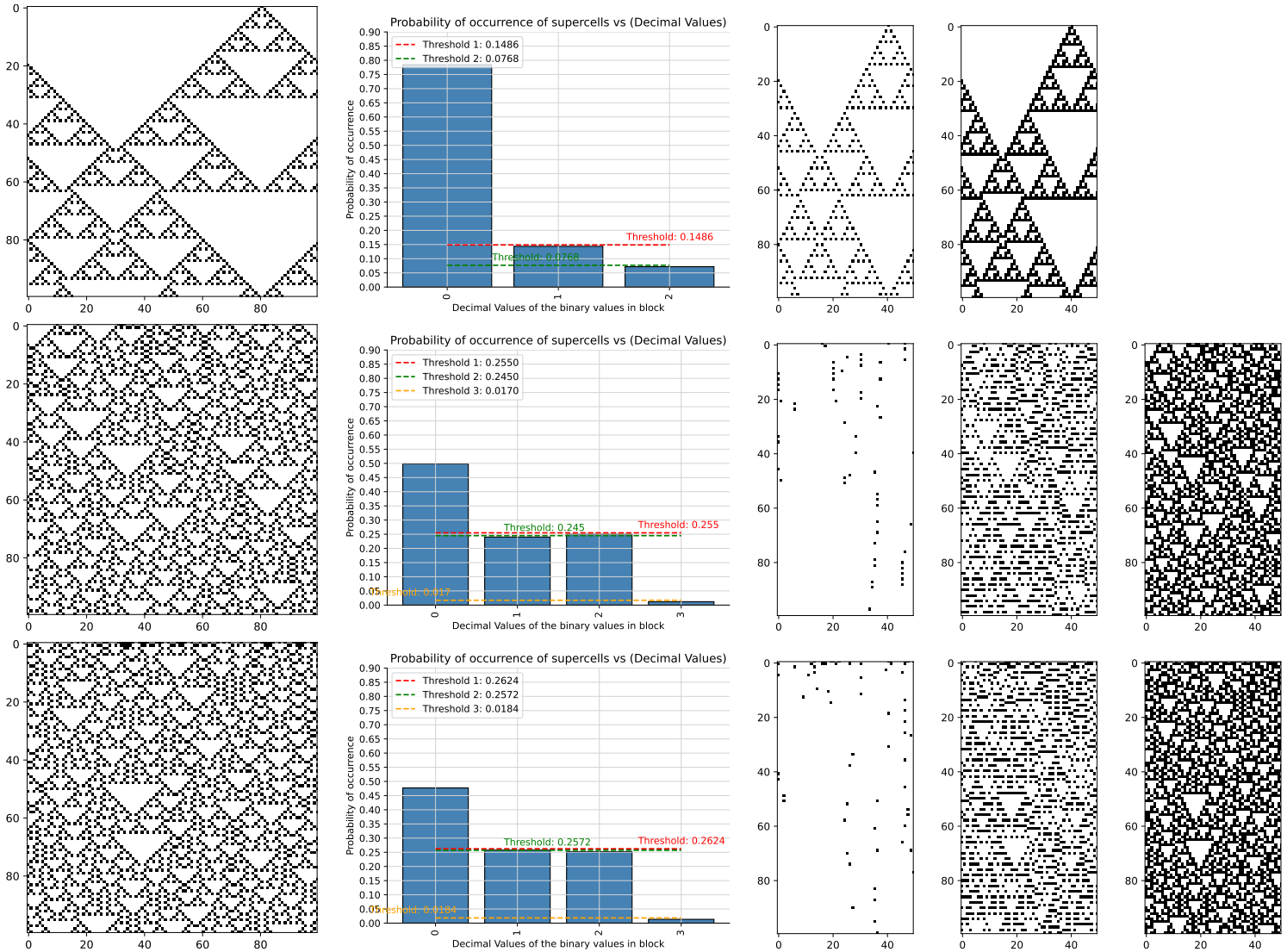

Table 74: FHCG plots for ECA Rule 150.

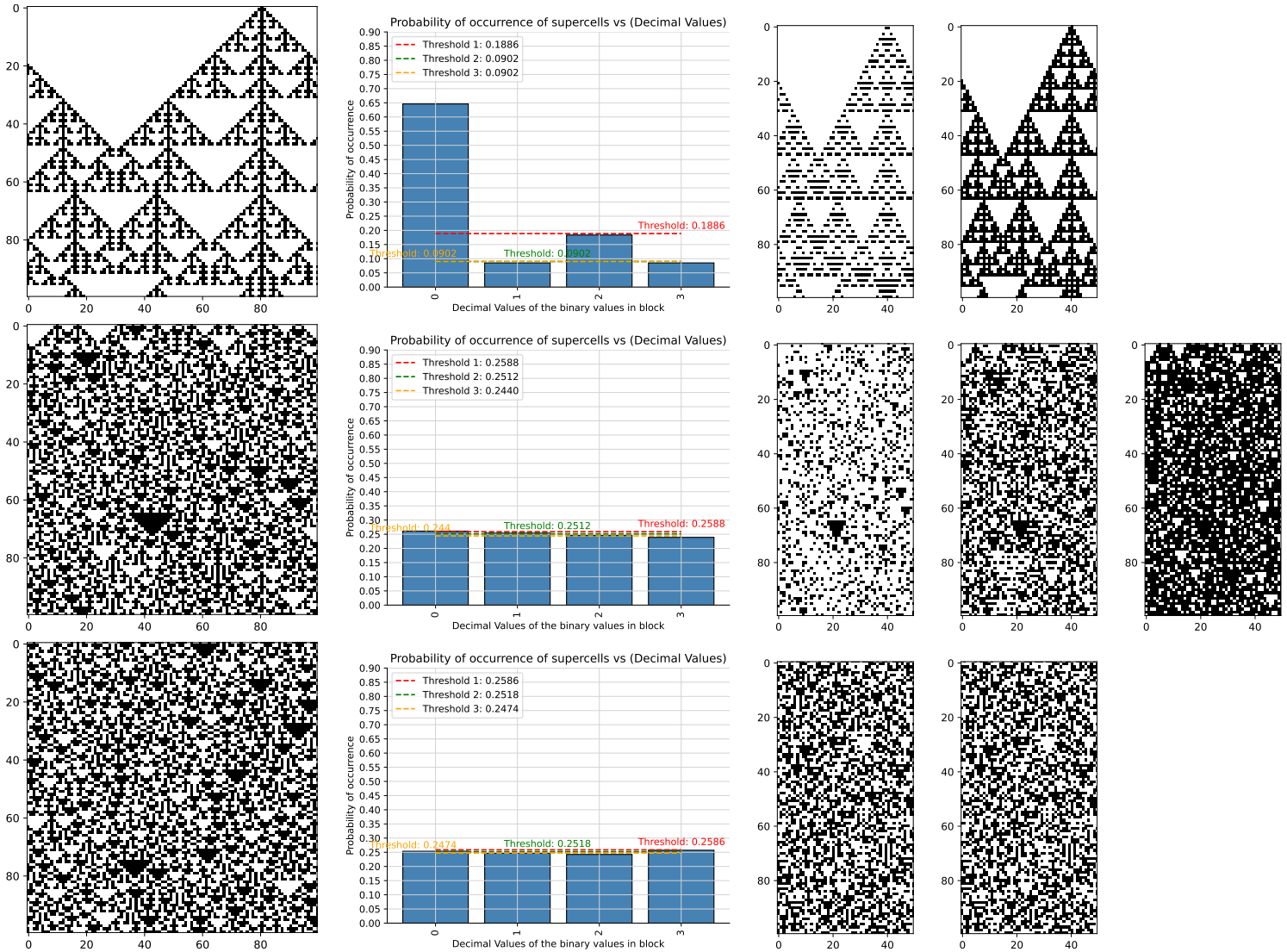

Table 75: FHCG plots for ECA Rule 152.

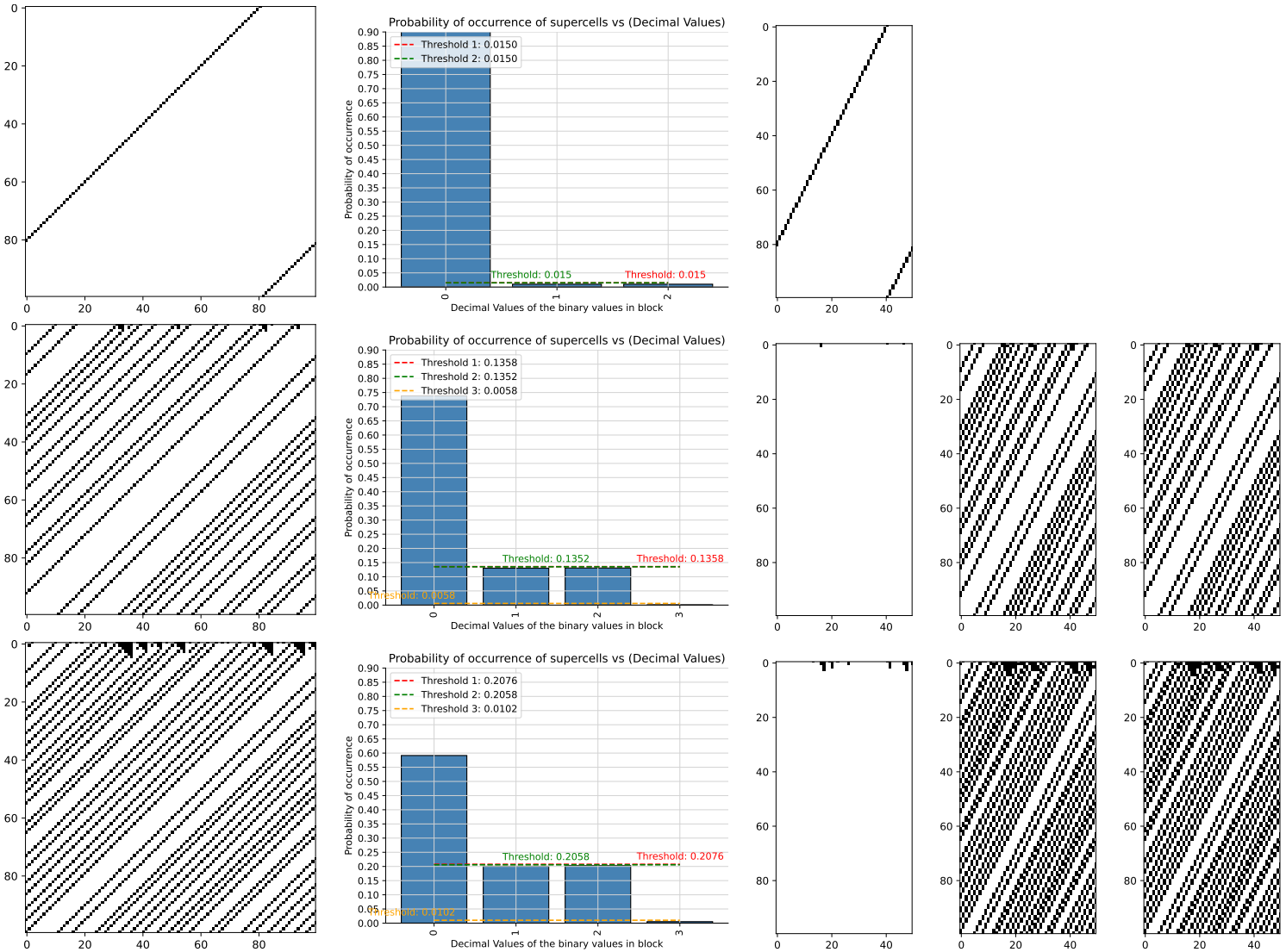

Table 76: FHCG plots for ECA Rule 154.

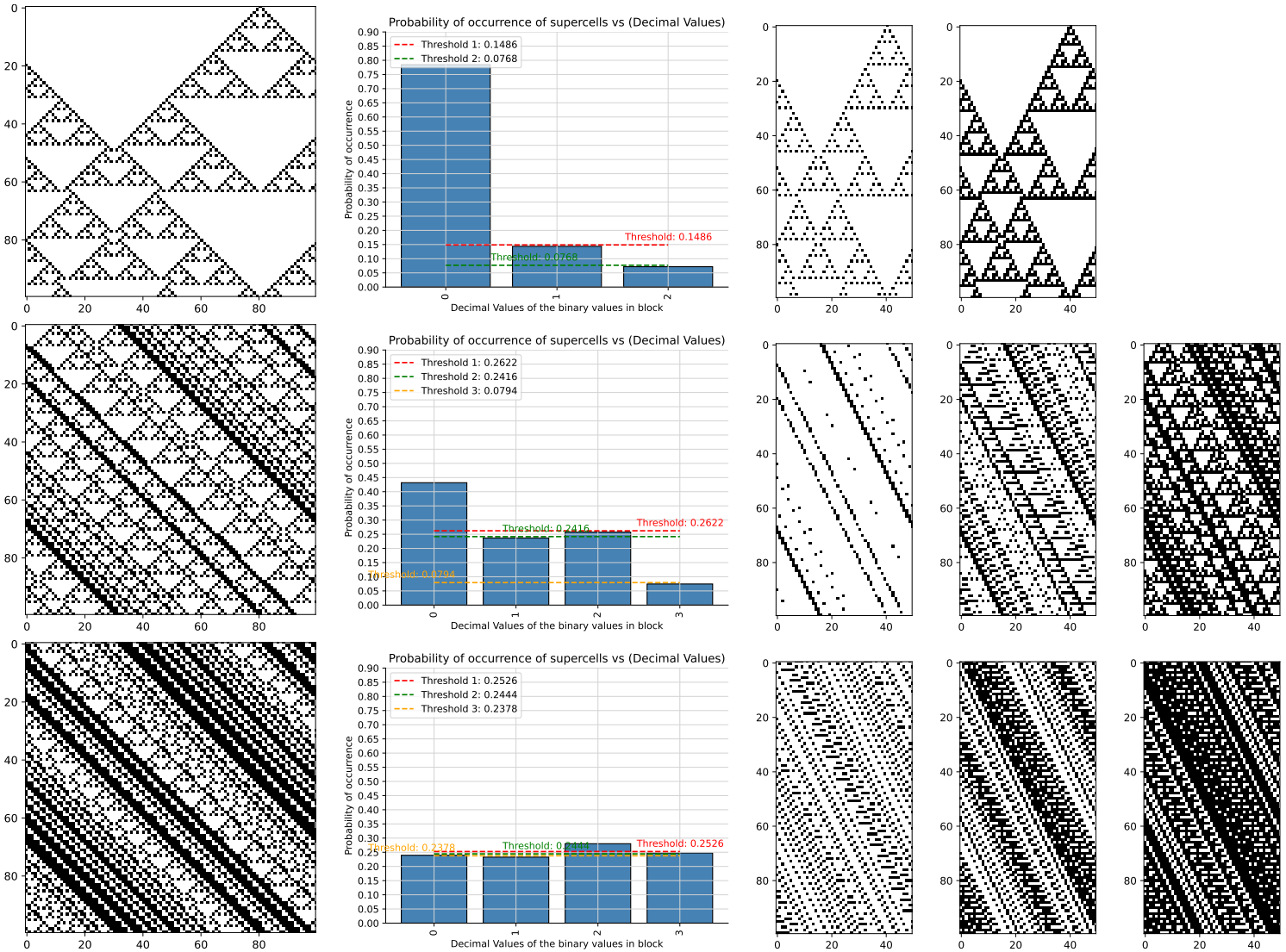

Table 77: FHCG plots for ECA Rule 156.

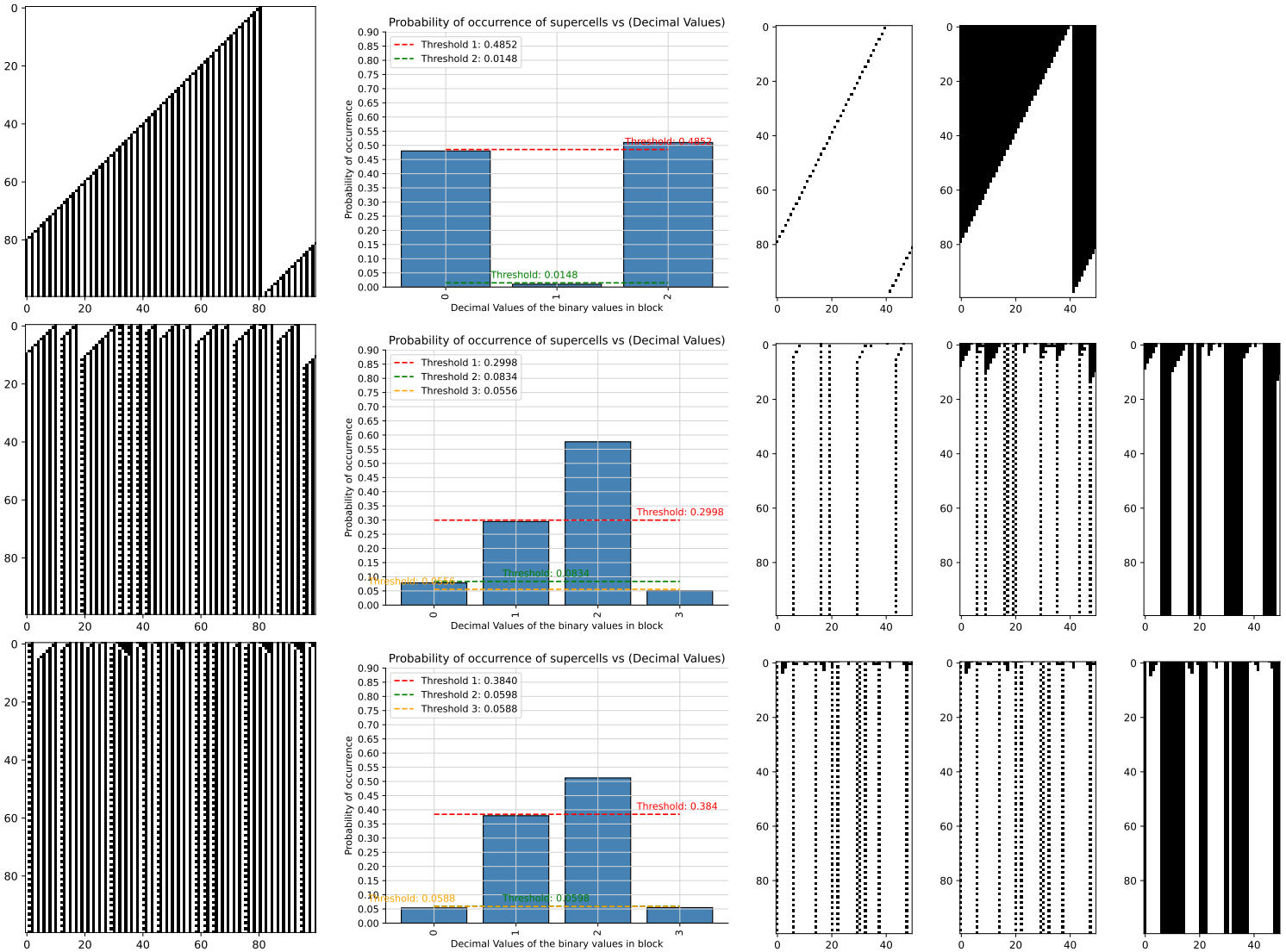

Table 78: FHCG plots for ECA Rule 160.

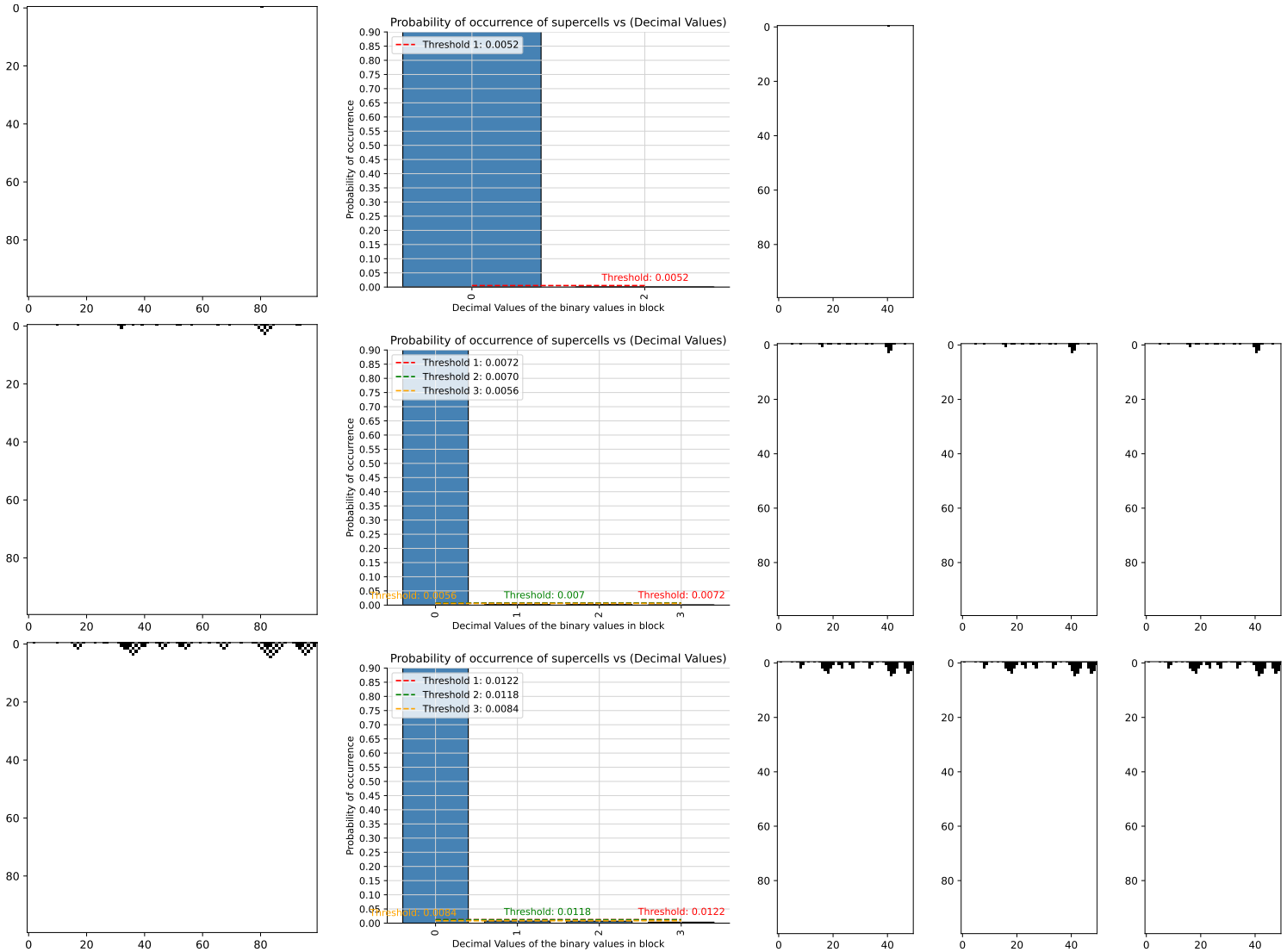

Table 79: FHCG plots for ECA Rule 162.

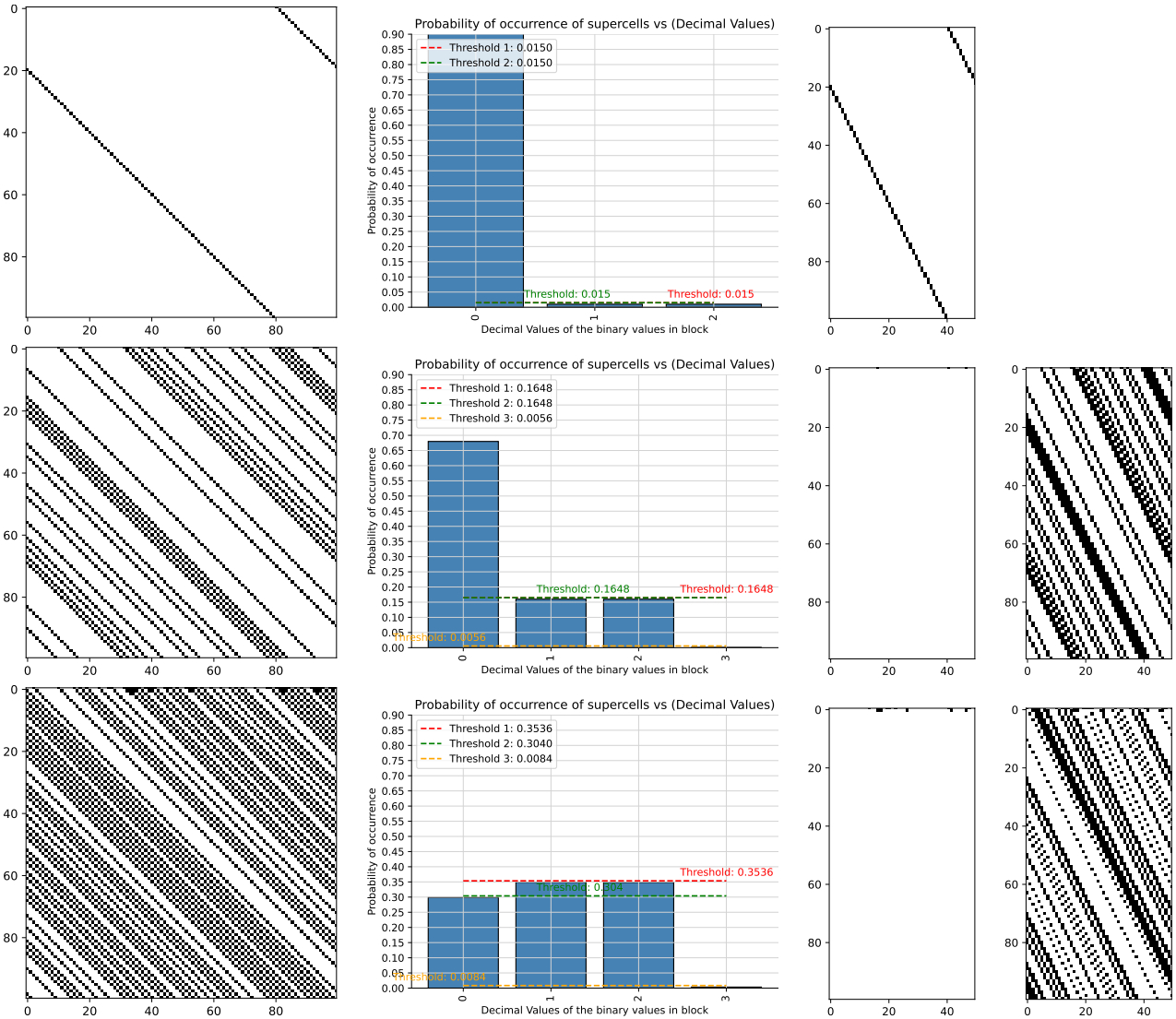

Table 80: FHCG plots for ECA Rule 164.

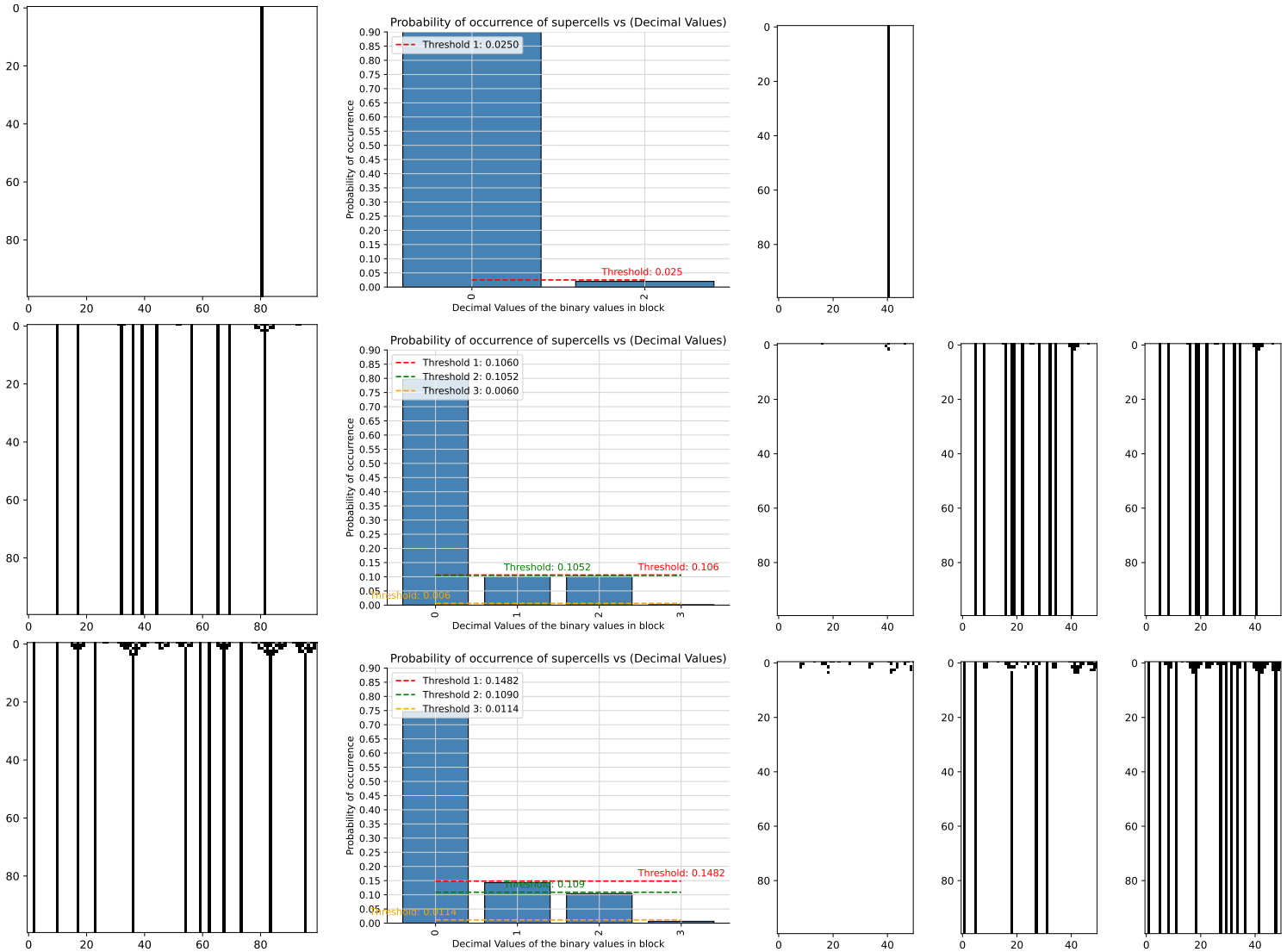

Table 81: FHCG plots for ECA Rule 168.

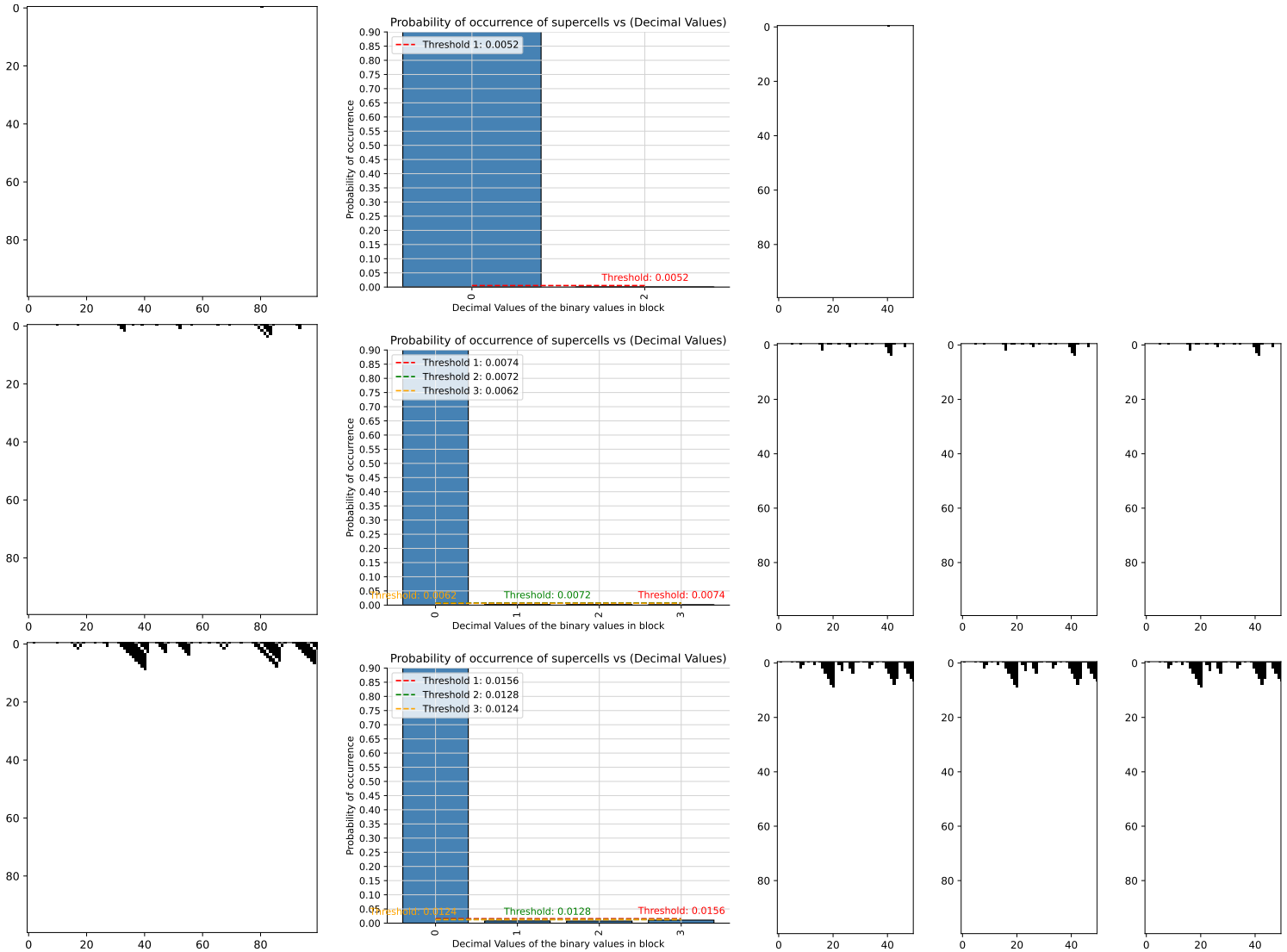

Table 82: FHCG plots for ECA Rule 170.

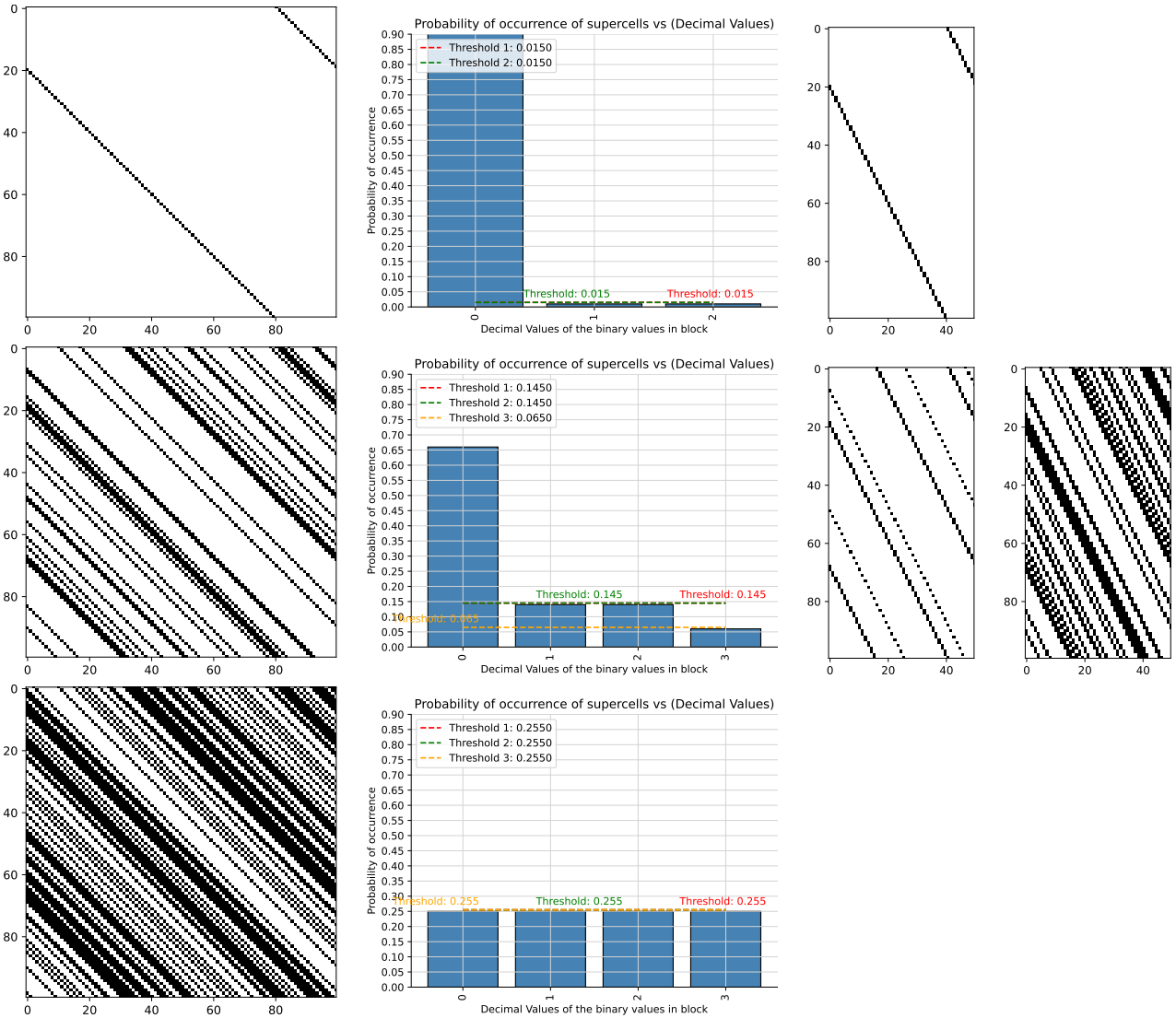

Table 83: FHCG plots for ECA Rule 172.

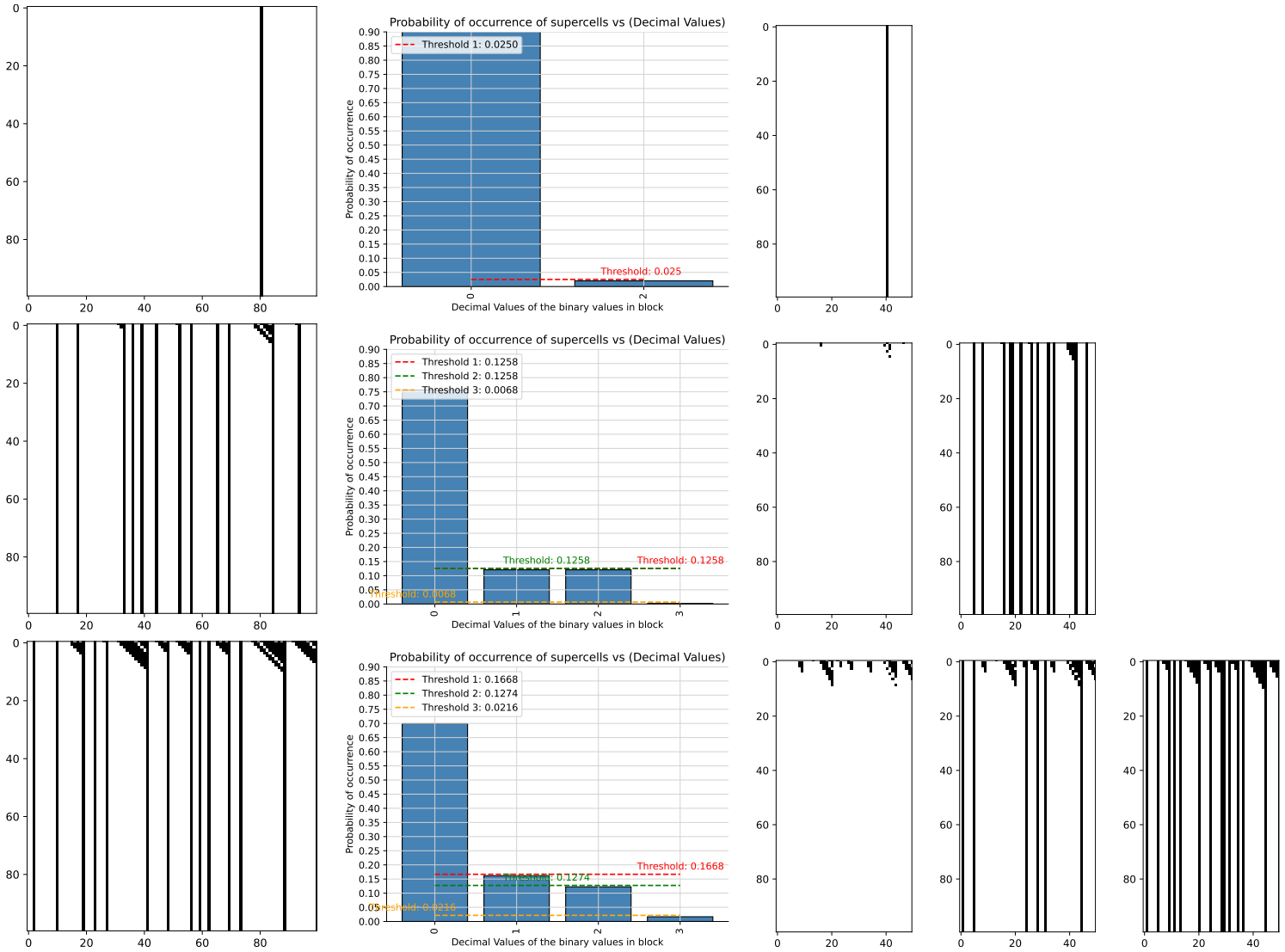

Table 84: FHCG plots for ECA Rule 178.

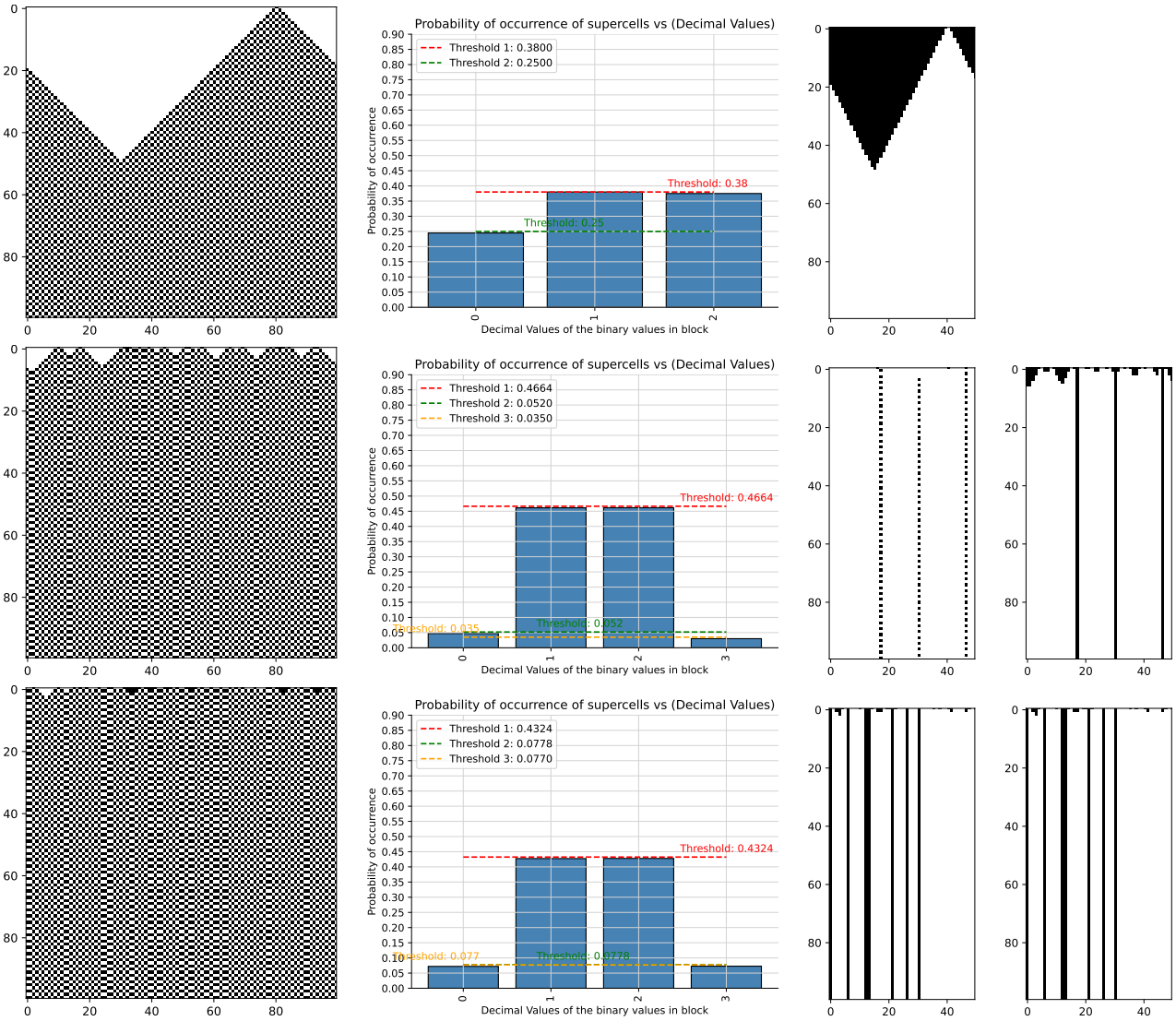

Table 85: FHCG plots for ECA Rule 184.

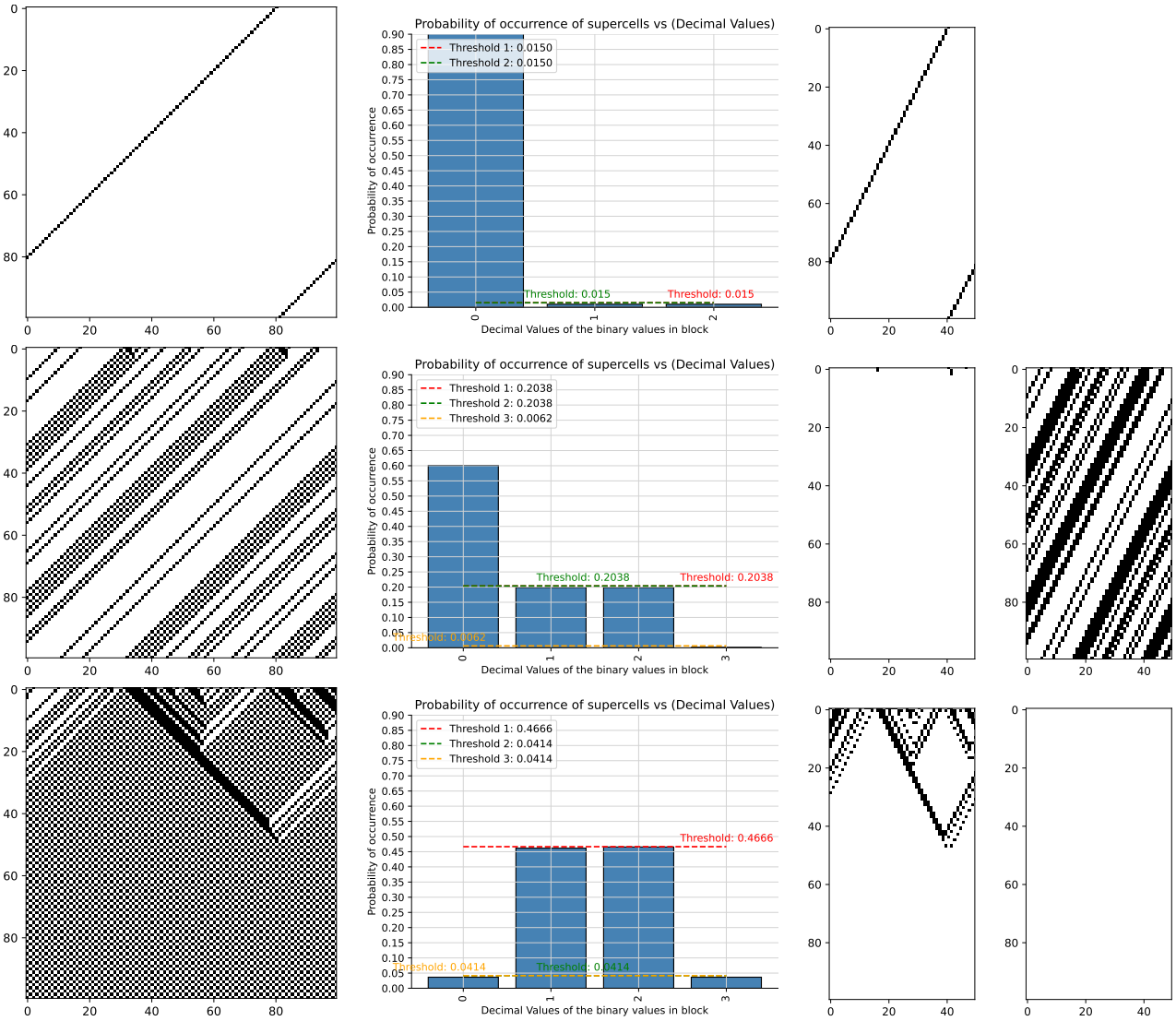

Table 86: FHCG plots for ECA Rule 200.

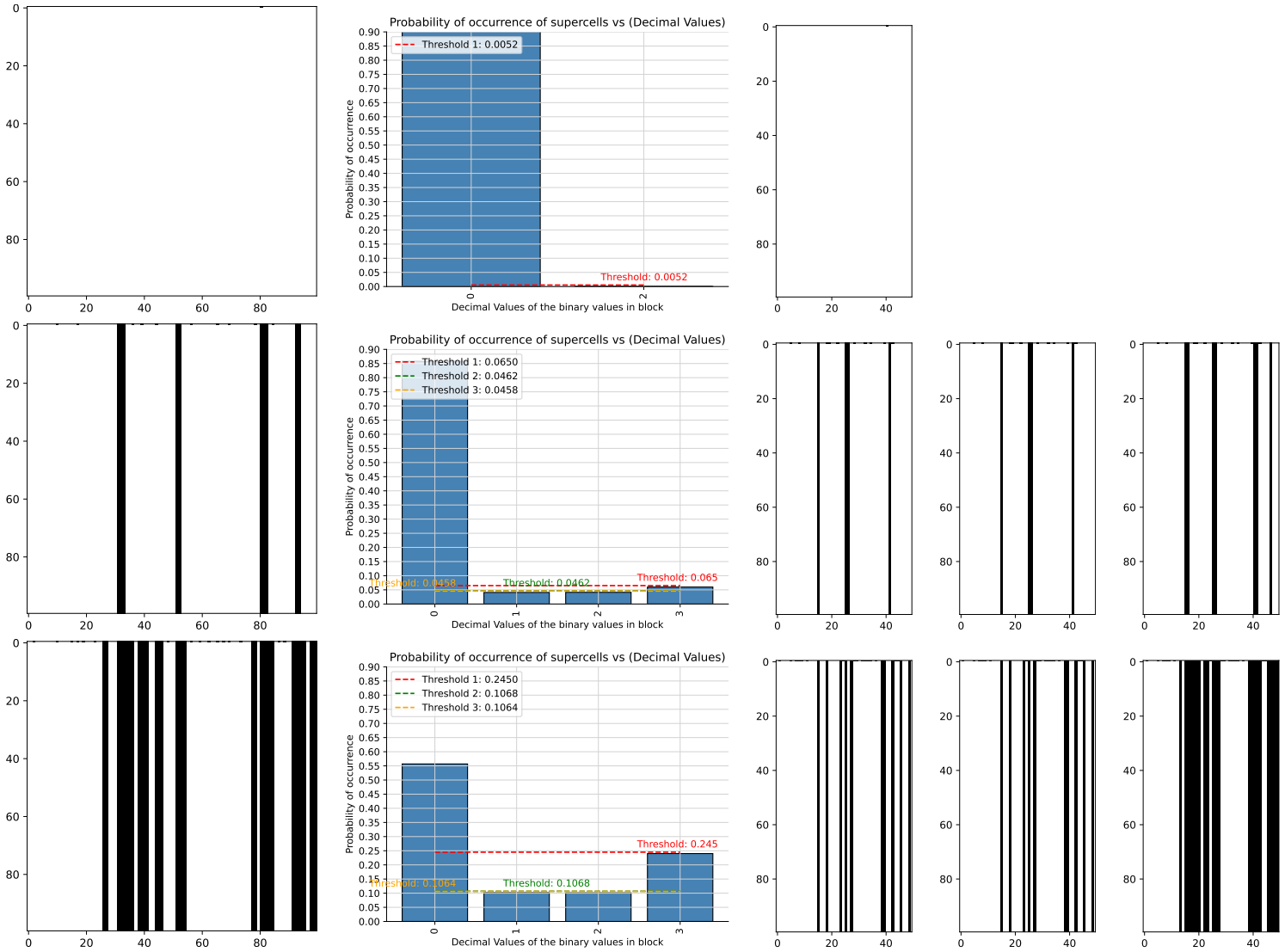

Table 87: FHCG plots for ECA Rule 204.

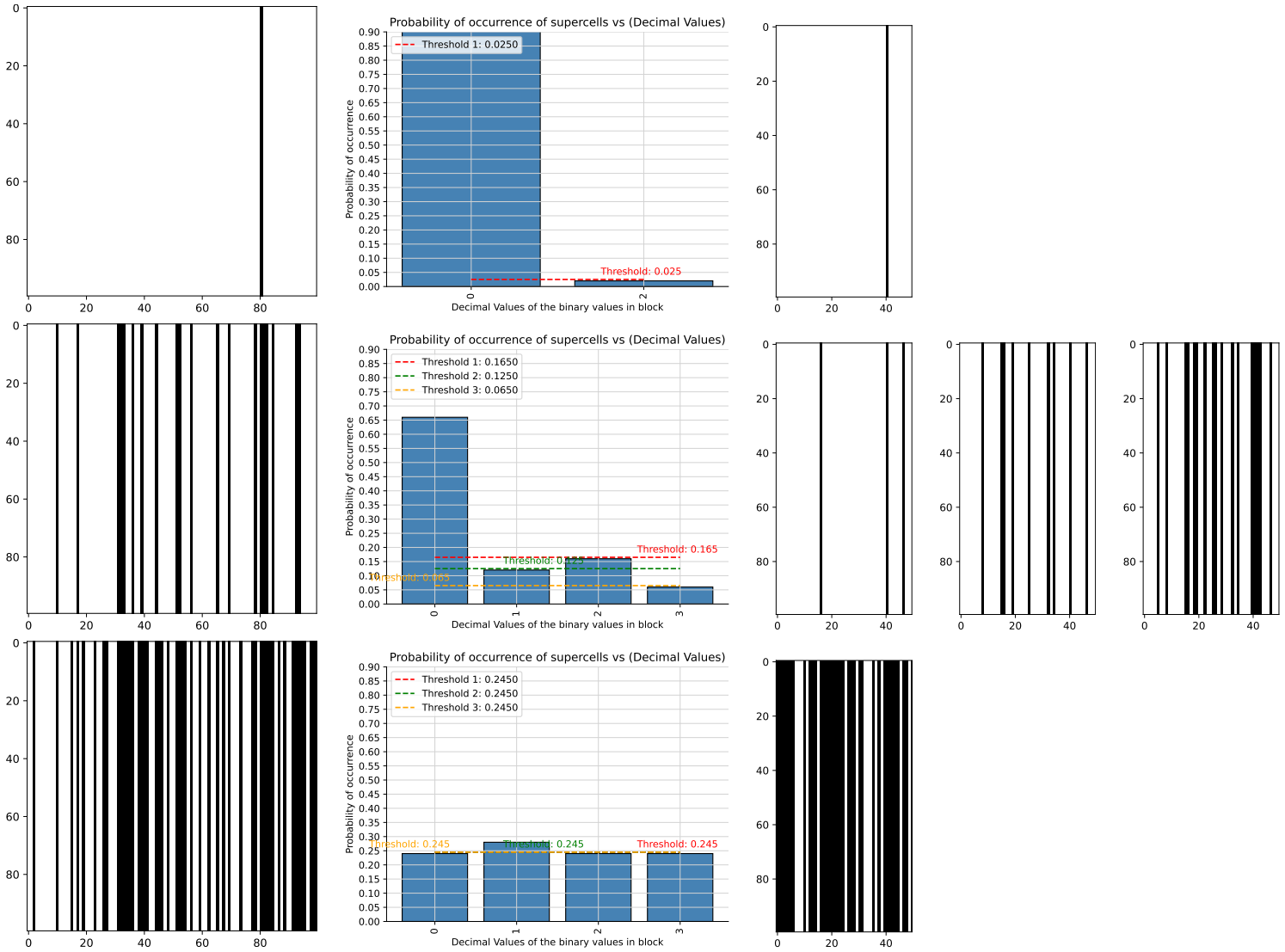

Table 88: FHCG plots for ECA Rule 232.

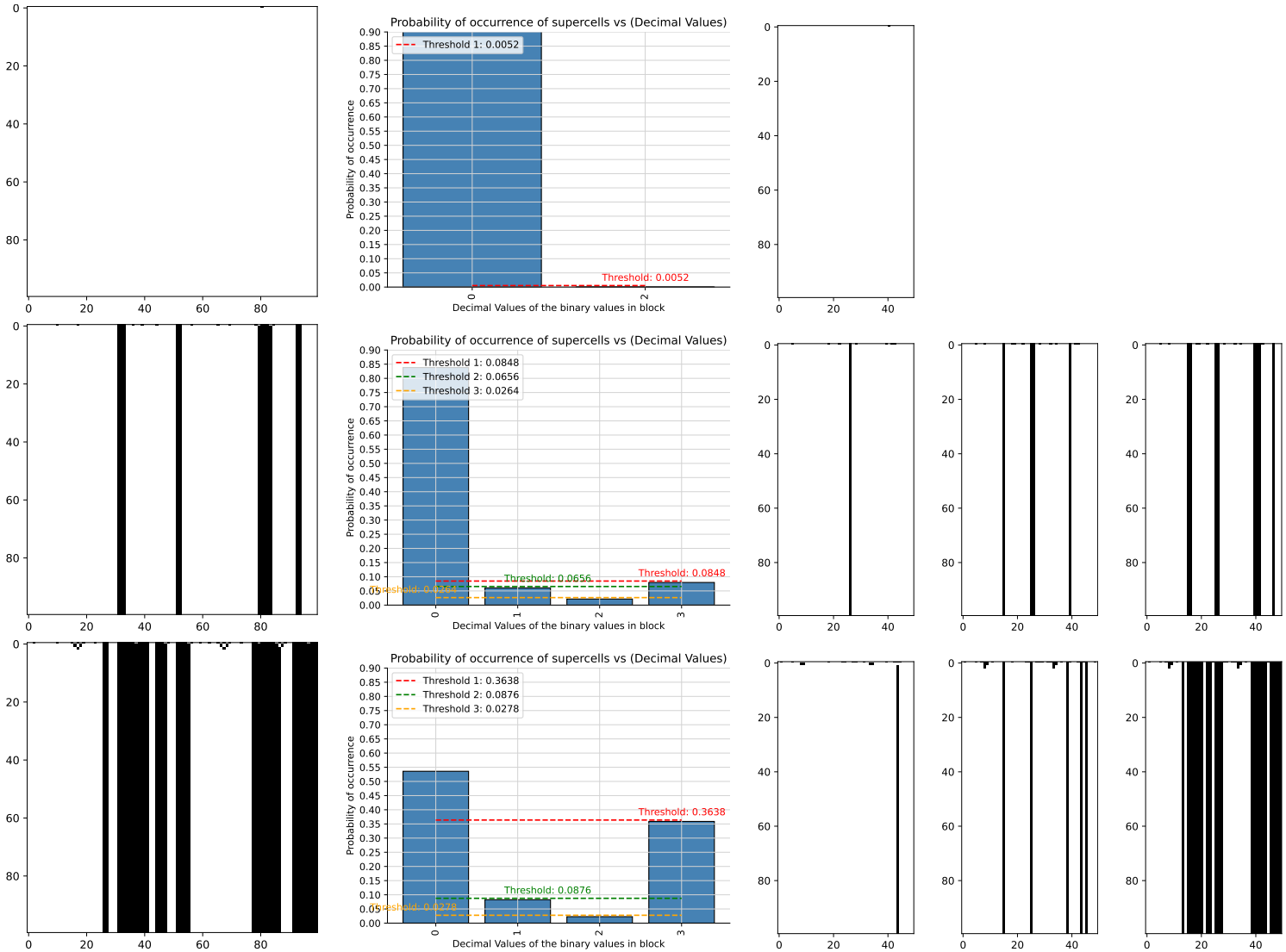

Table 89: Game of Life from MNCA with standard B3/S23 rule. The figure shows high definition (above) and FHCG (below). The chosen handpicked threshold for FHCG-Evo-MNCA is 0.011

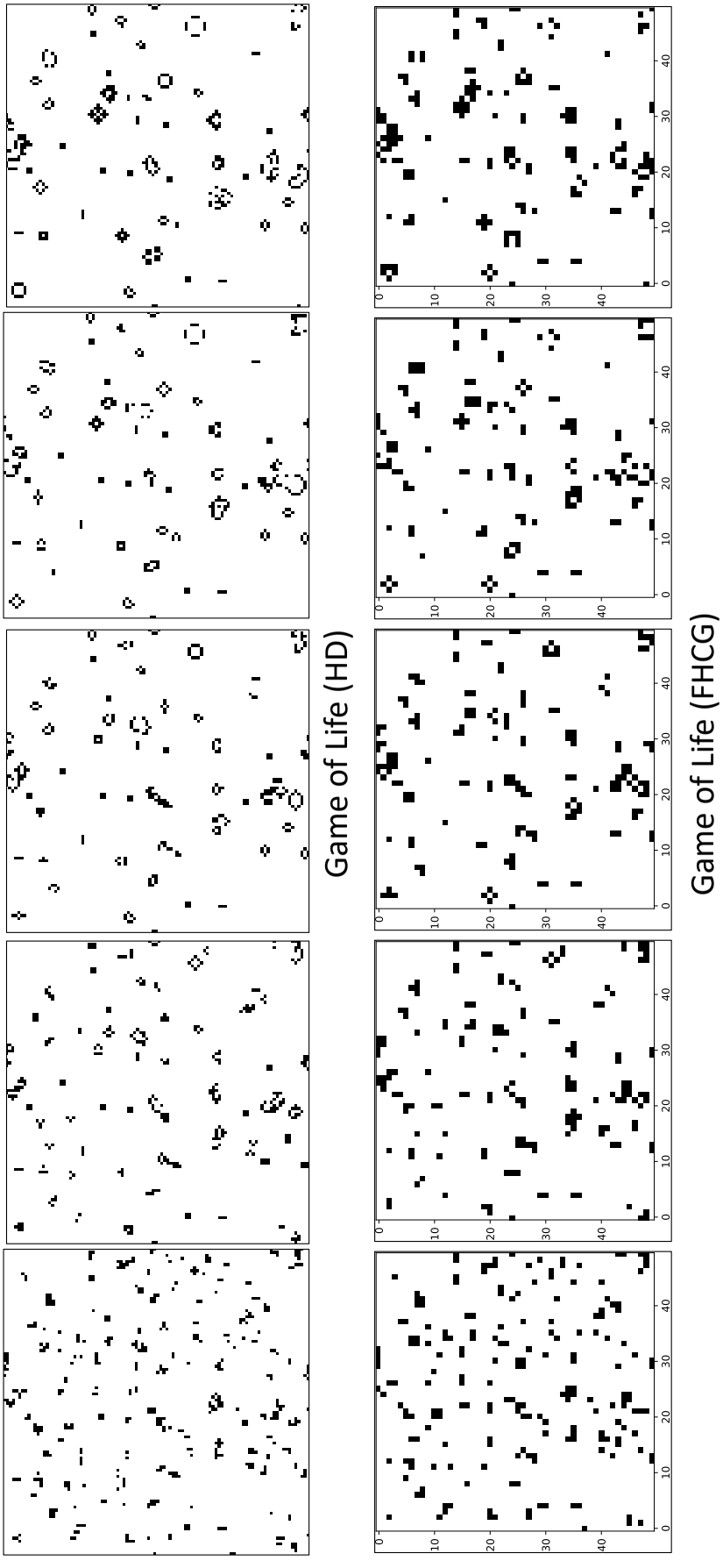

Table 90: Example of evolved rule from Evo-MNCA The values of the evolved genotype are shown on top of the image. The figure shows high definition (above) and FHCG (below). The chosen handpicked threshold for FHCG-Evo-MNCA is 0.025