ISOTROPIC CELLULAR AUTOMATA

the DDLab iso-rule paradigm

Andrew Wuensche

August 2020

To respect physics and nature, cellular automata models of self-organisation, emergence, computation and logical universality should be isotropic, having equivalent dynamics in all directions.

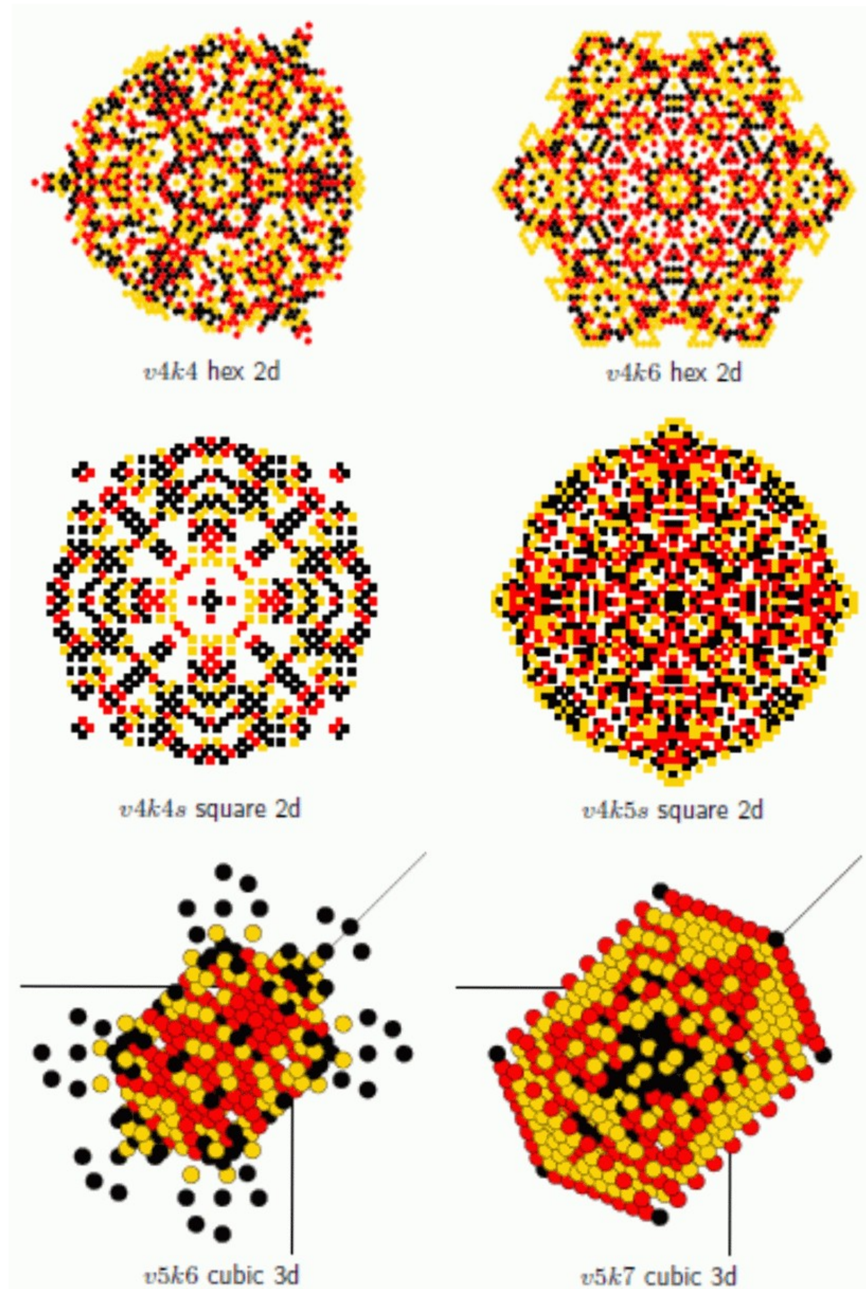
We present a novel paradigm, the iso-rule, a concise expression for isotropic cellular automata by the output table for each isotropic neighborhood group, allowing an efficient method of navigating and exploring iso-rule-space. Iso-groups and iso-rules apply for multi-value as well as binary, in one, two and three dimensions

Iso-rules provide an intermediate granularity between isotropic rules based on full lookup-tables, and isotropic subsets --- totalistic, reaction-diffusion and survival/birth rules.

With these methods its now possible to identify the critical iso-groups driving glider-gun/eater dynamics, find more examples, and search for underlying principles of self-organization.

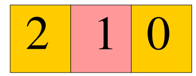
iso-rules: initial symmetry must be preserved

patterns from a singleton seed, v=4, 2d and 3d



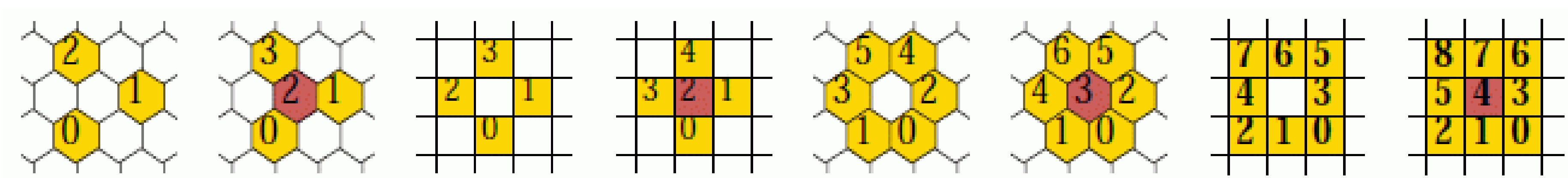
iso-rules apply to 1d, 2d and 3d neighborhood templates, for binary v=2, and multi-value v>=3

v=value-range (number of colors) k=neighborhood size

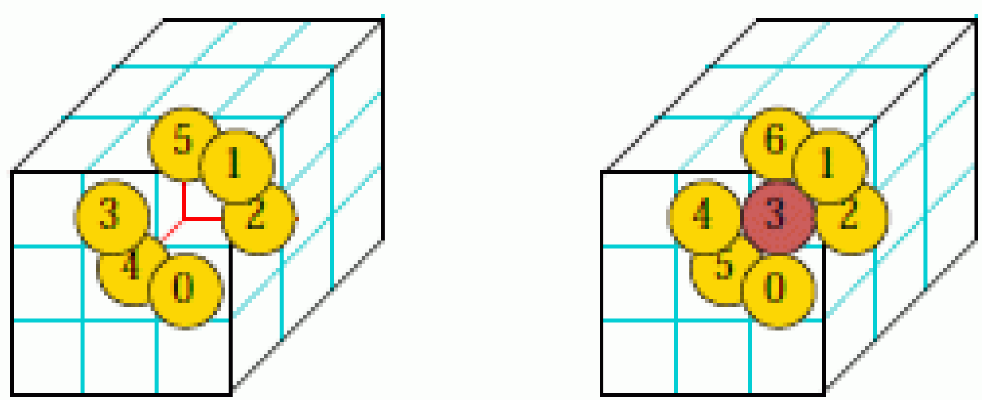


1d … any neighborhood size, k

2d : k=3 to k=9, k=4 can be either hex or square



3d: k=6 or k=7



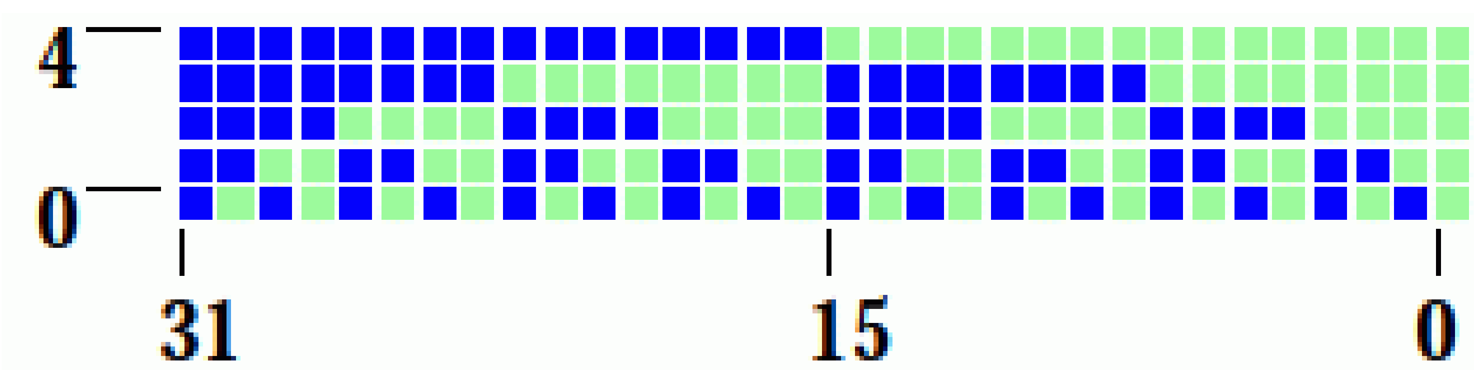
template geometry is chosen for best symmetry – target cell sometimes included

full lookup tables – complete list of v^k neigborhoods – rcode

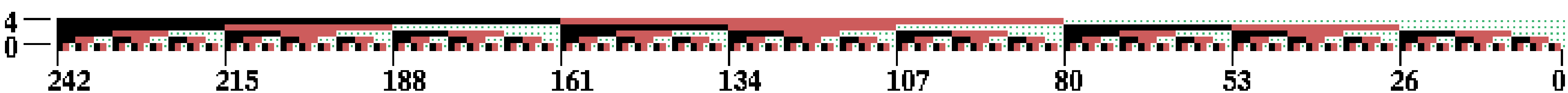
follows template indexing – independent of template geometry

Wolfram's convention

```
 7    6    5    4    3    2    1    0    - binary value and rcode index
111  110  101  100  011  010  001  000   - k=3 neighborhoods
 0    0    1    1    1    1    0    0    - output string = rcode 60 in decimal
```



$2^5=32$



$3^5=243$

isotropic subsets of rule types

isotropic by default but can be re-expressed as rcode to transform to iso-rule

totalistic rules

tcode

```
20....................0 - all possible totals
 |                    |
 444433332222111100000 - tcode, outputs [0,1,2,3,4]
                         (hex) 49236da491248000
```

$$S_t = k(v-1)+1.$$

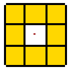
kcode

```
                  27..........................0 <--k=6 kcode index
                    |                          |
                > 2: 6554443333222221111110000000  < frequency strings       6    0
v=3 values      > 1: 0102103210432105432106543210  < of 2s, 1s, 0s,      from 0 to 0
                > 0: 0010120123012340123450123456  < shown vertically        0    6
                     ||||||||||||||||||||||||||||
                     0022000220022001122200021210 <--kcode, outputs [0,1,2]
                                    beehive rule (hex) 0a0282816a0264
```

outer-totalistic

0: 000001000 - birth: exactly 3 live neighbors
1: 000001100 - survival: 2 or 3 live neighbors

S_o is $v \times S_k$ for kcode, or $v \times S_t$ for tcode.

 $k8$ template

$$S_k = (v+k-1)!/(k! \times (v-1)!)$$

for binary: tcode=kcode

reaction-diffusion

```
   resting(0)--->if within t
          \                \
           \              (1)  excited
            \               \
             (v-2)<----(3)<--(2)  refractory
```

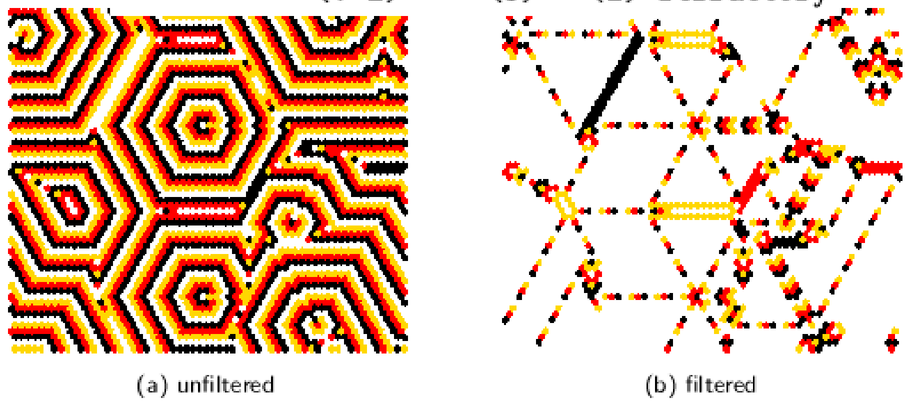


Figure 14: Snapshots of a hex 2d reaction-diffusion CA $v4k7$, with the iso-rule (size 1720) shown below. The threshold interval was set 1 to 4. The initial state 60×60 has a low density (0.01) of non-zero cells. (a) the emergent pattern, and (b) the pattern with the 3 most frequent iso-groups filtered, showing structures that resemble glider-guns.

survival/birth

game-of-Life s23/b3 – rcode > iso-rule

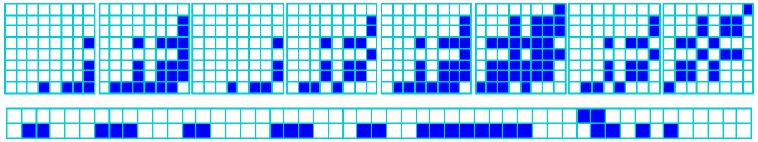
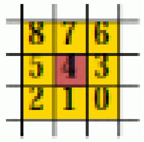


Figure 15: Conway's game-of-Life (s23/b3) shown as a 512 bit rcode in 8 rows. The diagonal symmetry in each 8x8 block is a necessary (but insufficient) indication of isotropy but a useful visual clue for the general case of isotropic rcode for a binary $v2k9$ 2d CA with a Moore neighborhood. Below the rcode is the 102 bit iso-rule — (hex) 00 00 00 00 00 60 03 1c 61 c6 7f 86 a0.

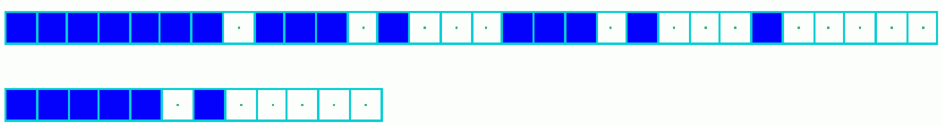


binary Moore template alternatives to iso-rule
Hensel notation for Golly – DDLab compatible
Emmauel Sapin's notation from his publications

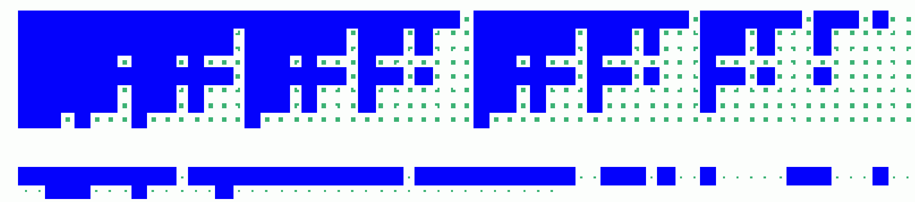
algorithm transforms full lookup-table to isotropic rcode

(examples are majority rules)

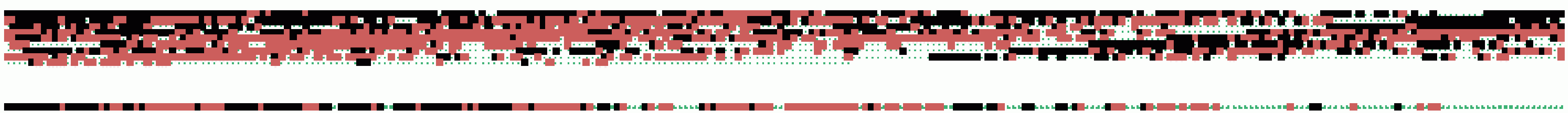
then rcode to iso-rule



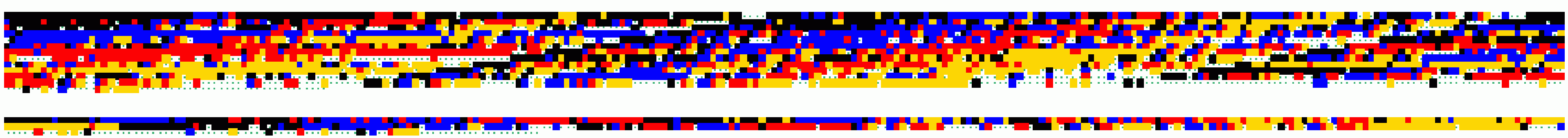
(a) $v2k5$ rcode(32), 2d square iso-rule(12)



(b) $v2k9$ rcode(512), 2d square iso-rule(102)



(c) $v3k7$ rcode(2187), 2d hex iso-rule(276)



(d) $v5k5$ rcode(3225), 2d square iso-rule(600) the sizes of iso-rules for 2d and 3d are difficult to calculate analytically. The tables below give iso-rule sizes (number of iso-groups) computed algorithmically –- much shorter than rcode v^k

2d hex-k

v \ k	3	4	6	7
2	4	8	13	26
3	10	30	92	276
4	20	80	430	1720
5	35	175	1505	
6	56	336		
7	74	588		
8	120	960		

2d square-k

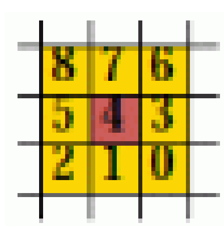
v \ k	4	5	8	9
2	6	12	51	102
3	21	63	954	2862
4	55	220		
5	120	600		
6	231	1386		
7	406			
8	666			

3d

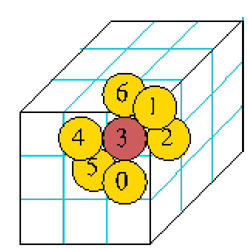
v \ k	6	7
2	10	20
3	57	171
4	240	960
5	800	

note: rcode sizes for...

2d hex
v3k7=276
Spiral rule

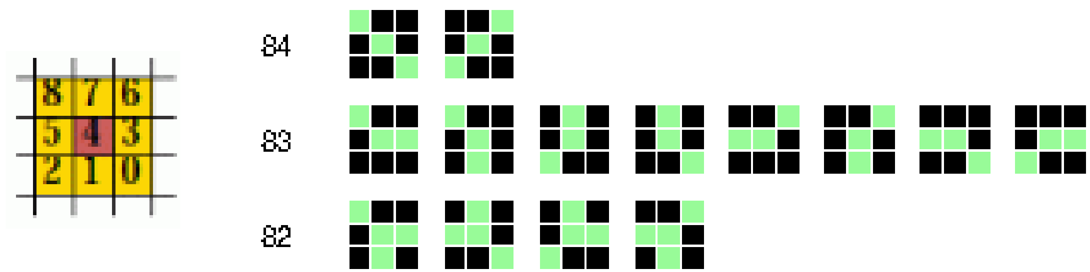


Moore 2d square
v2k9 =102
Game-of-Life

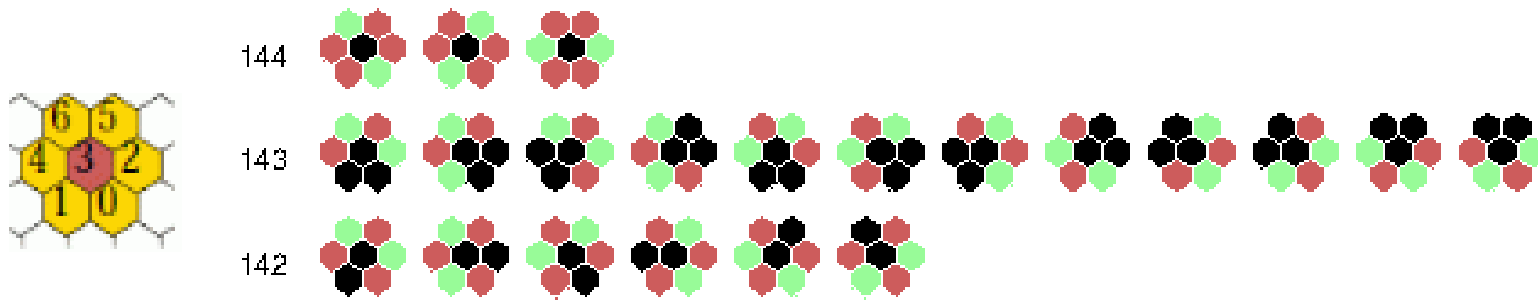


3d square cubic
v3k7 =171
Spiral rule

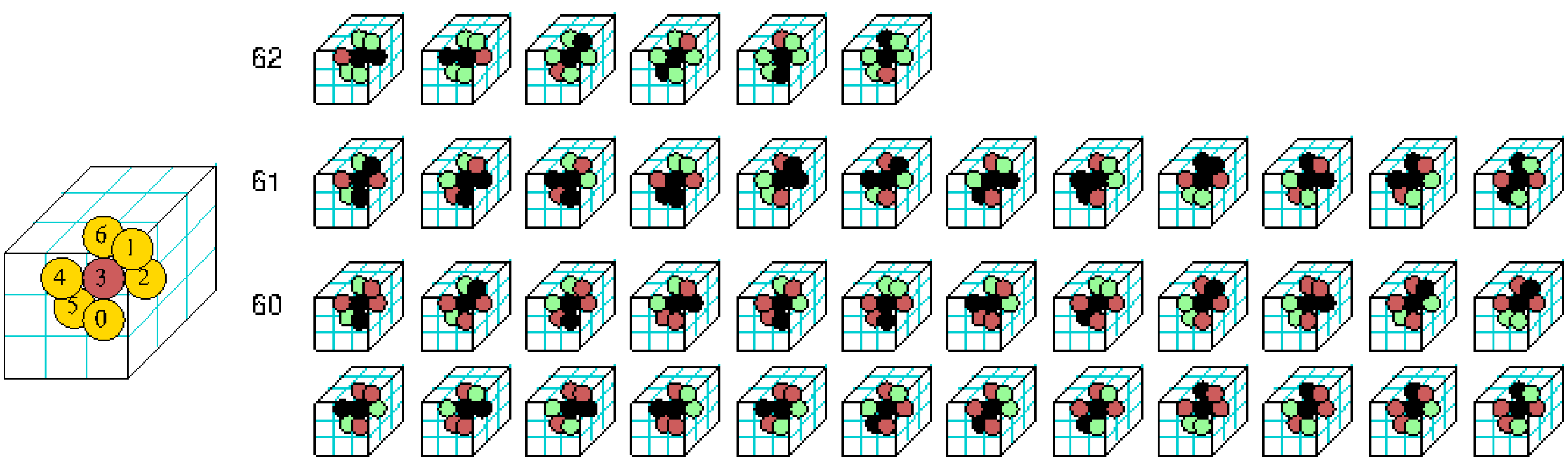
examples of iso-groups, represented by the left prototype



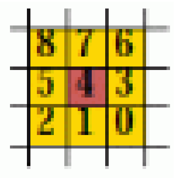
(a) 3 sucessive $v2k9$ 2d square (Moore) groups for iso-indeces as shown (max=101)



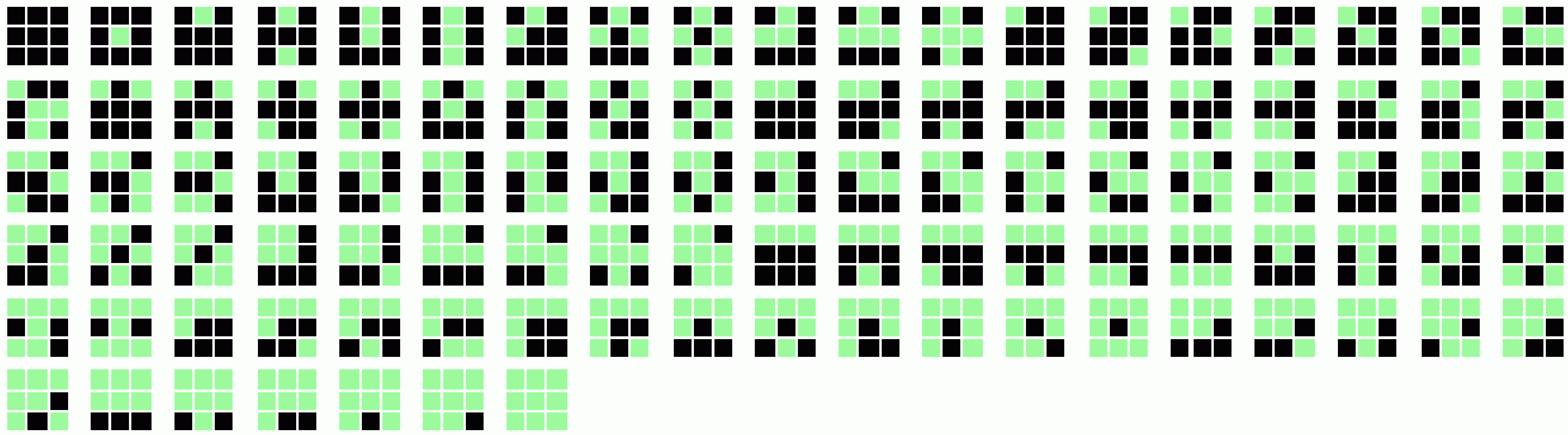
(b) 3 sucessive $v3k7$ 2d hex groups for iso-indeces as shown (max=275)



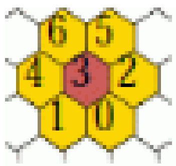
(c) 3 sucessive $v3k7$ 3d groups for iso-indeces as shown (max=171)



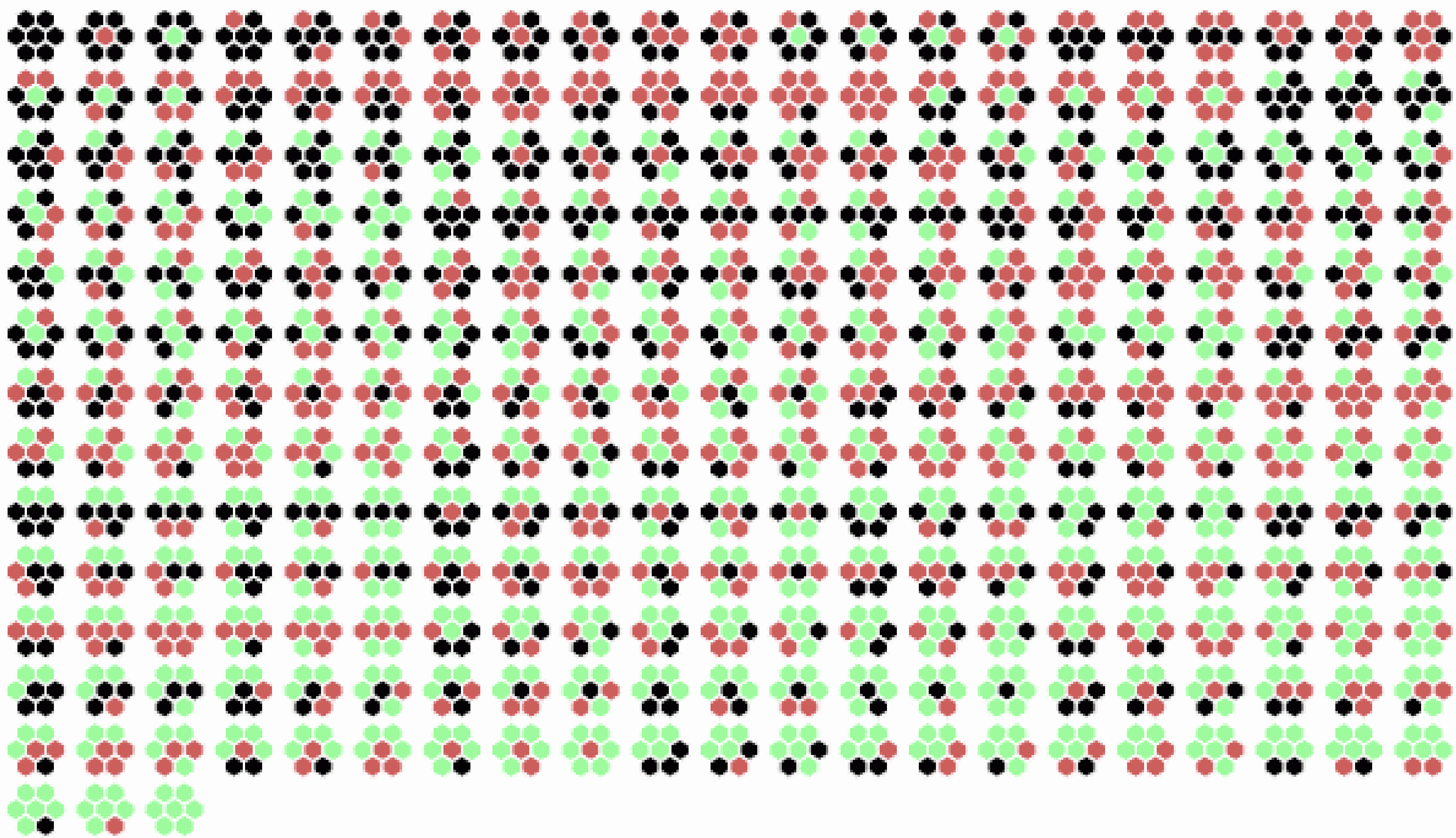
102 v2k9 2d iso-rule prototypes
each represents an iso-group



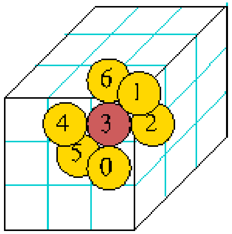
(a) 102 $v2k9$ 2d square (Moore) neighborhood iso-group prototypes



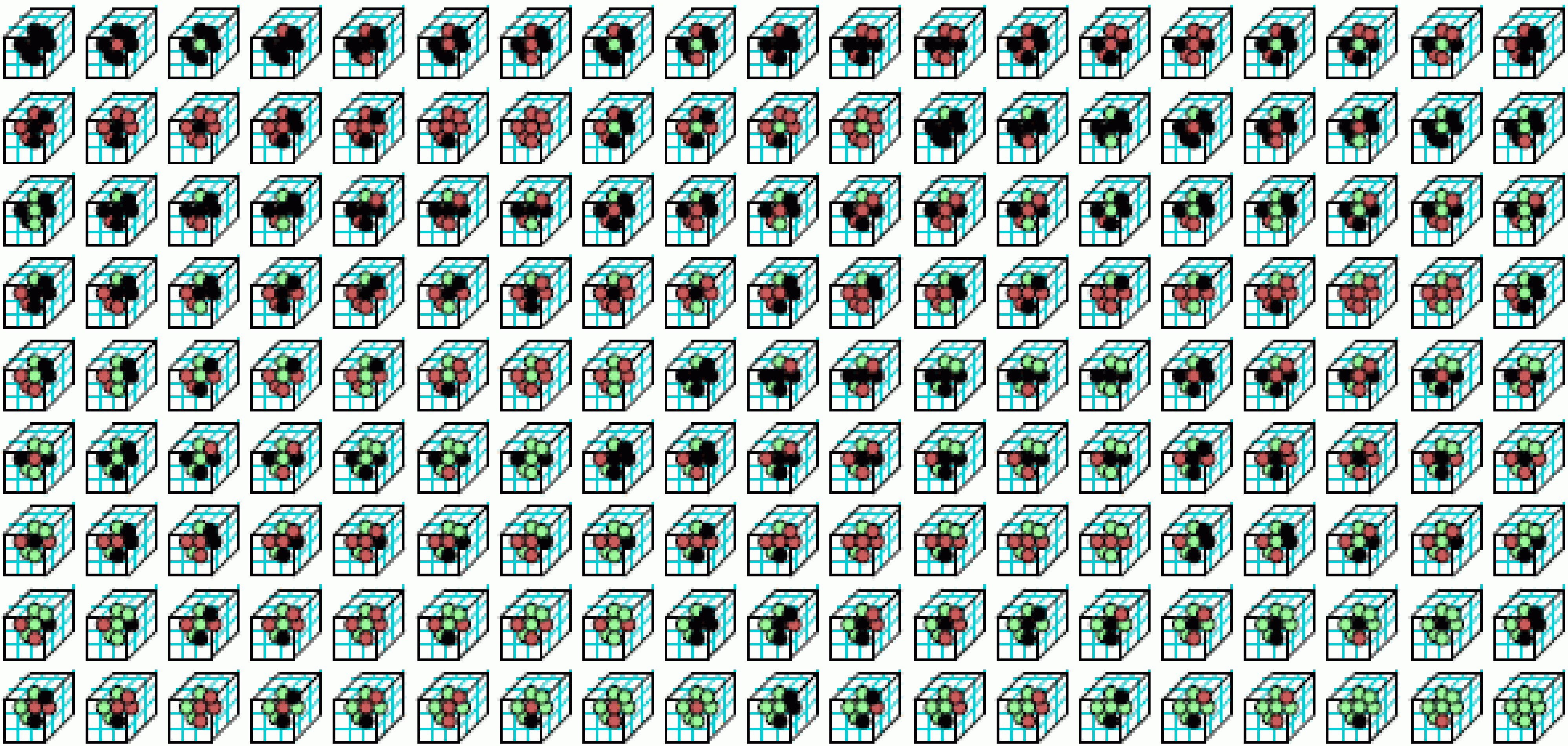
276 v3k7 2d hex iso-rule prototypes
each represents an iso-group



(b) 276 *v3k7* 2d hex neighborhood iso-group prototypes



172 v3k7 3d iso-rule prototypes
each represents an iso-group

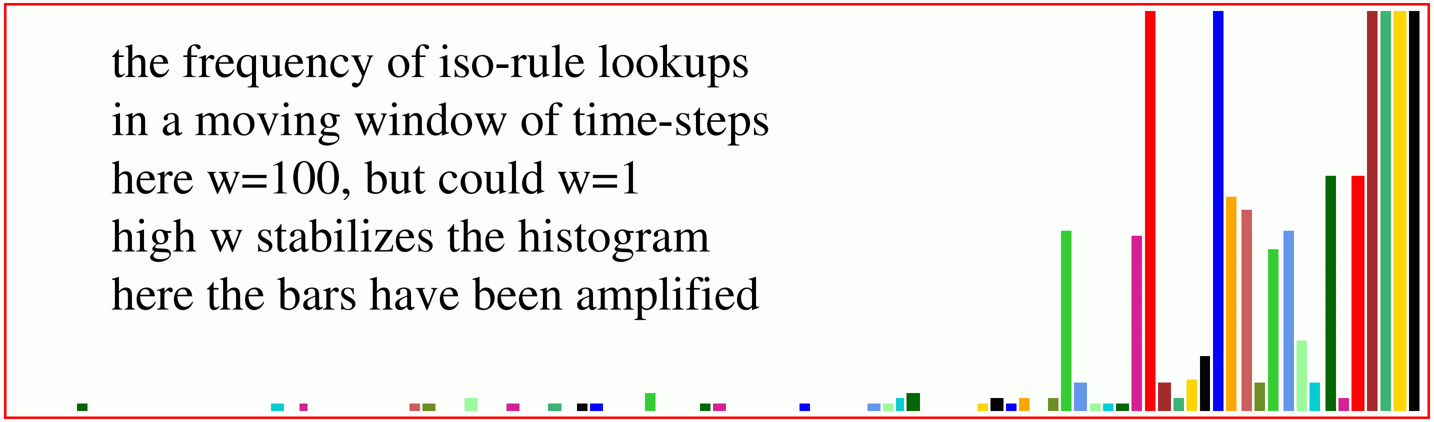


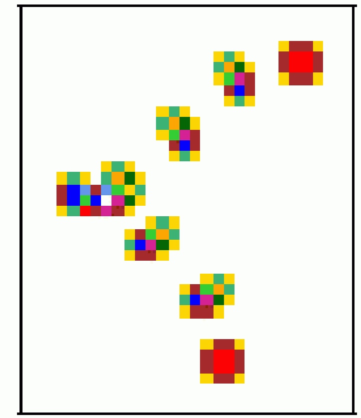
(c) 172 *v3k7* 3d neighborhood iso-group prototypes

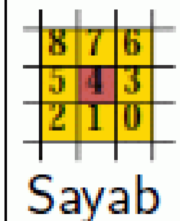
iso-rule input-frequency histogram

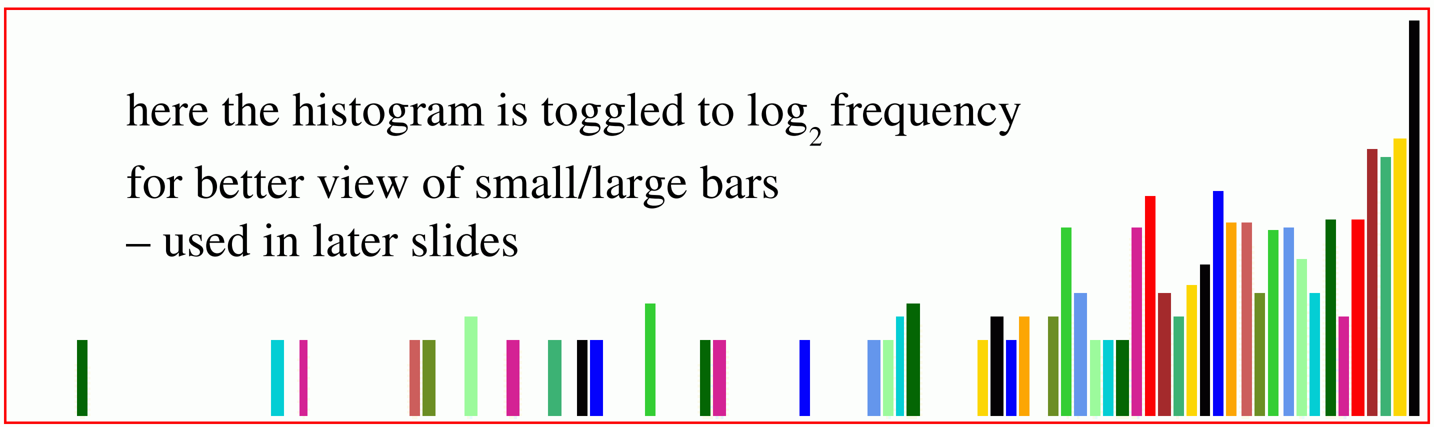
this and related functions are are present in DDLab for rcode, kcode, tcode -- now extended to iso-rules

this example: v2k9 Sayab rule – emergent glider-gun







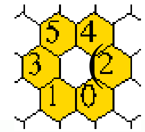


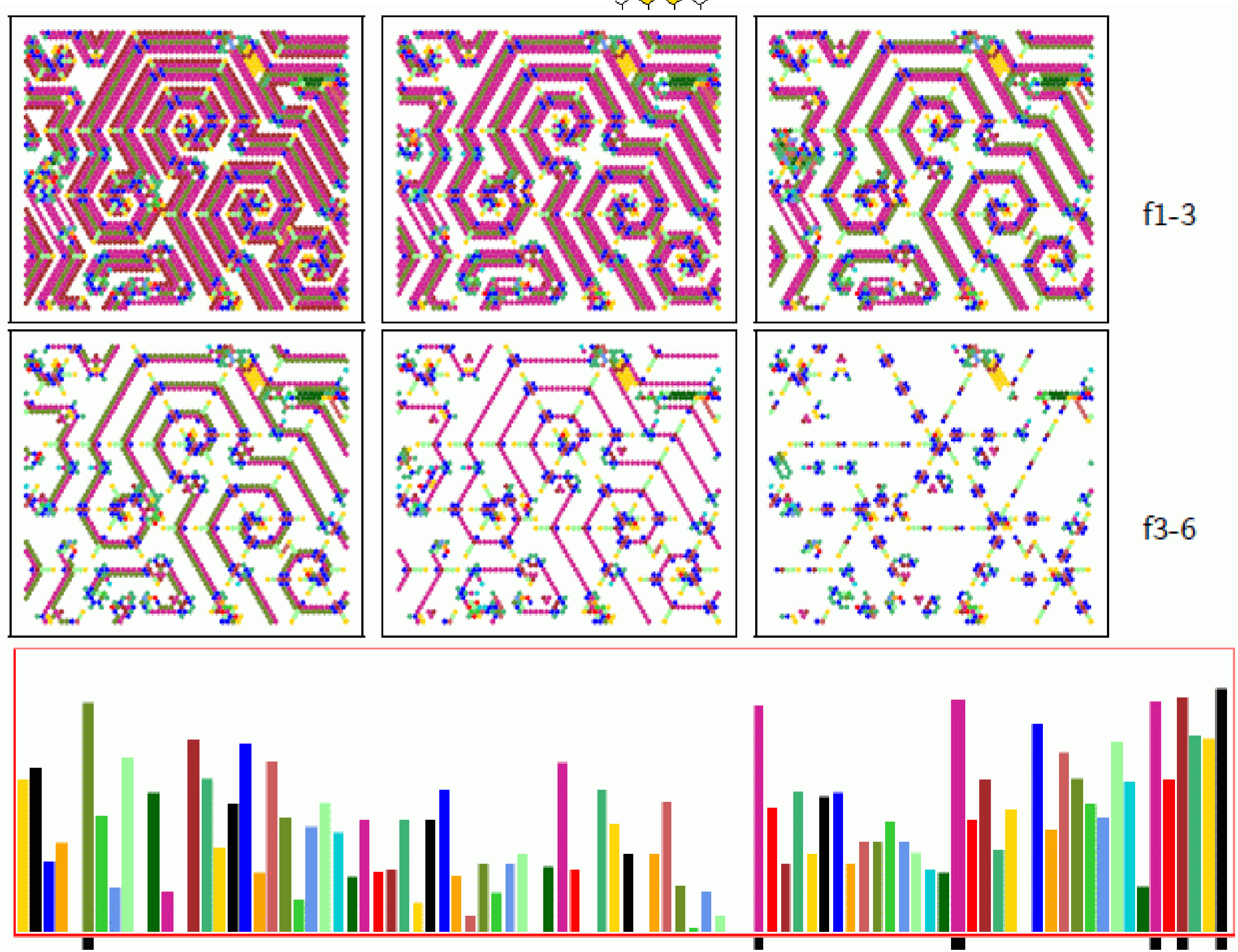
left: the same Sayab histogram as above but on the alternative $\log_2$ plot

entropy of the histogram (from the frequency plot) – (normalized 0-1) input-entropy

The Shannon entropy of the input-frequency histogram (the actual plot, not $\log_2$) measures its heterogeneity. The input-entropy H, at time-step t, for one time-step (w=1), is given by $H^t = -\sum_{i=0}^{S-1} (Q_i^t/n \times log_2 (Q_i^t/n))$, where Q_i^t is the lookup frequency of neighborhood i at time t. S is the rule-table size and n is the CA lattice size. The normalised entropy H_N is a value between 0 and 1, $H_N = H^t/log_2 n$, used in the graphic display and is usually averaged over a small moving window of time-steps.

On-the-fly iso-rule progressive filter (high to low – black block) unfilter (low to high)

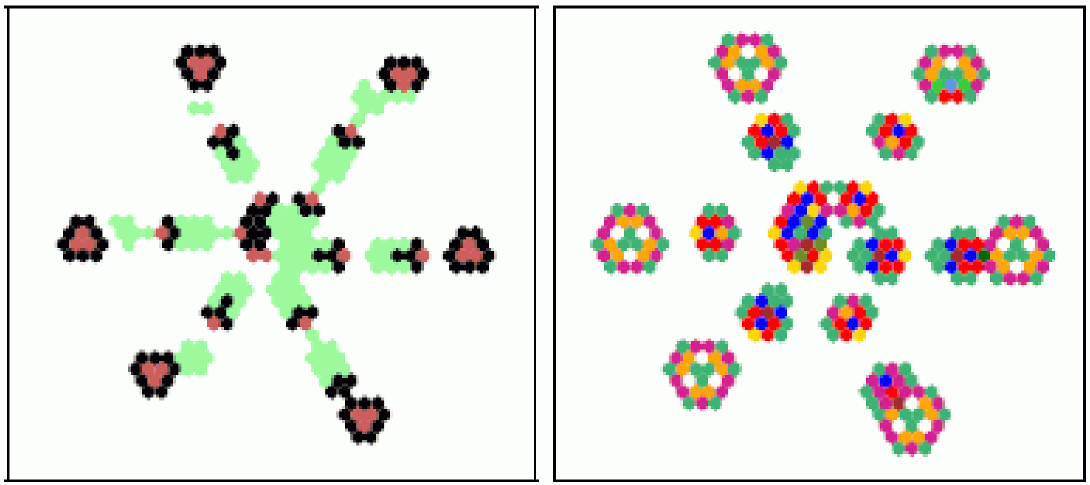
v3k6 hex  – 92 bars – emergent spirals

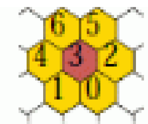


iso-rule interactive mutation

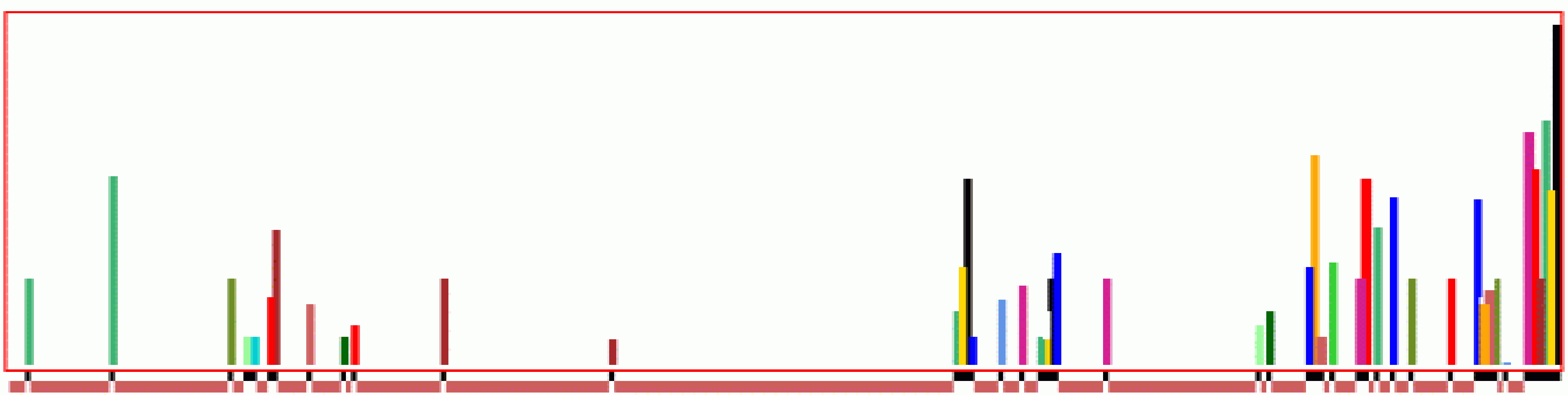
on-the fly random mutations aim for unfiltered bars first – and restored in reverse order while watching the effects on space-time patterns in a sort of mutation game

v3k7 spiral rule – hex lattice - emergent glider-guns





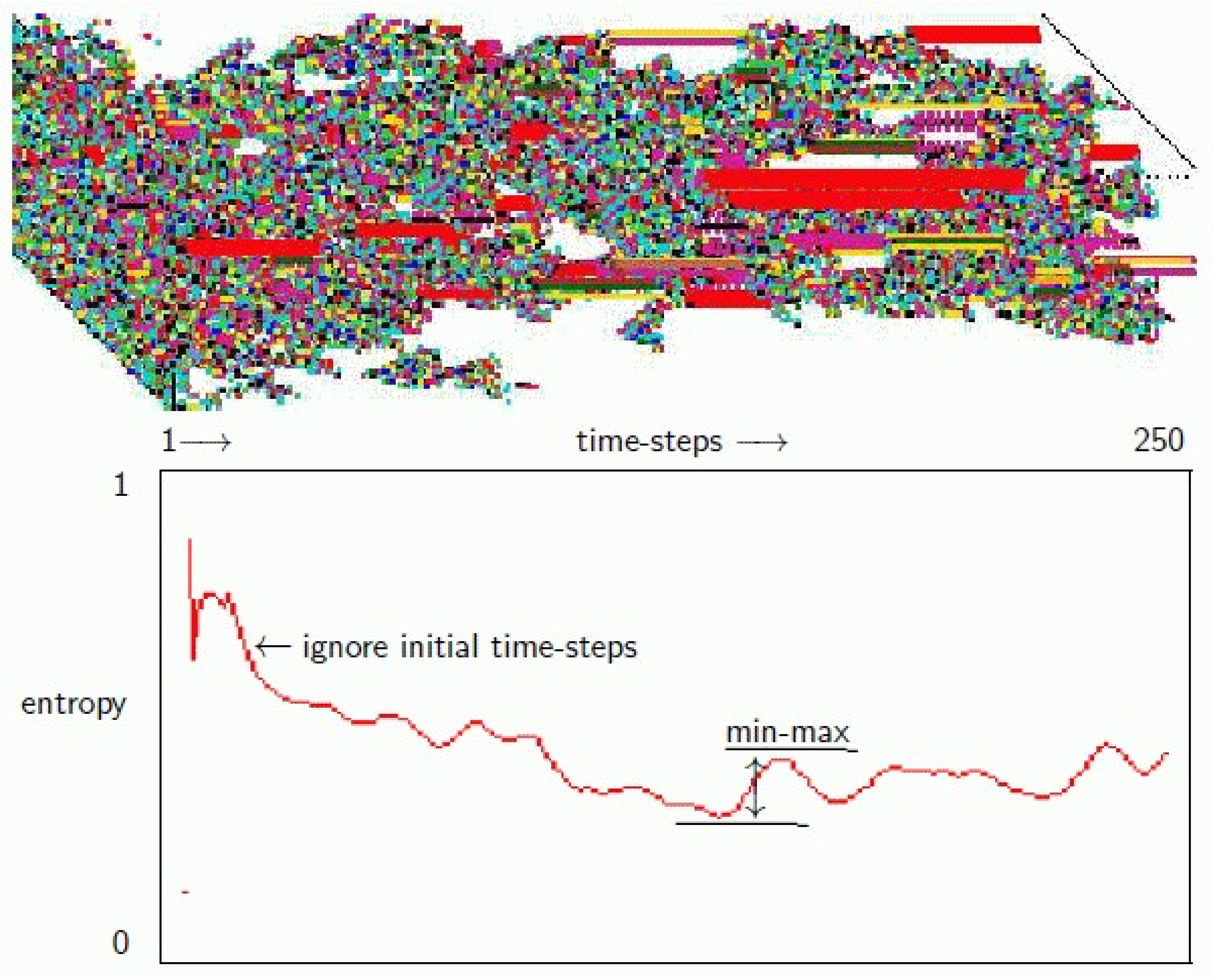
The Spiral glider-gun.
far left: colors by value with green dynamic trails.
near left: colors corresponding to histogram colors.



above: the iso-histogram representing 276 iso-groups, 46 are active. The iso-rules (hex) are compared below:

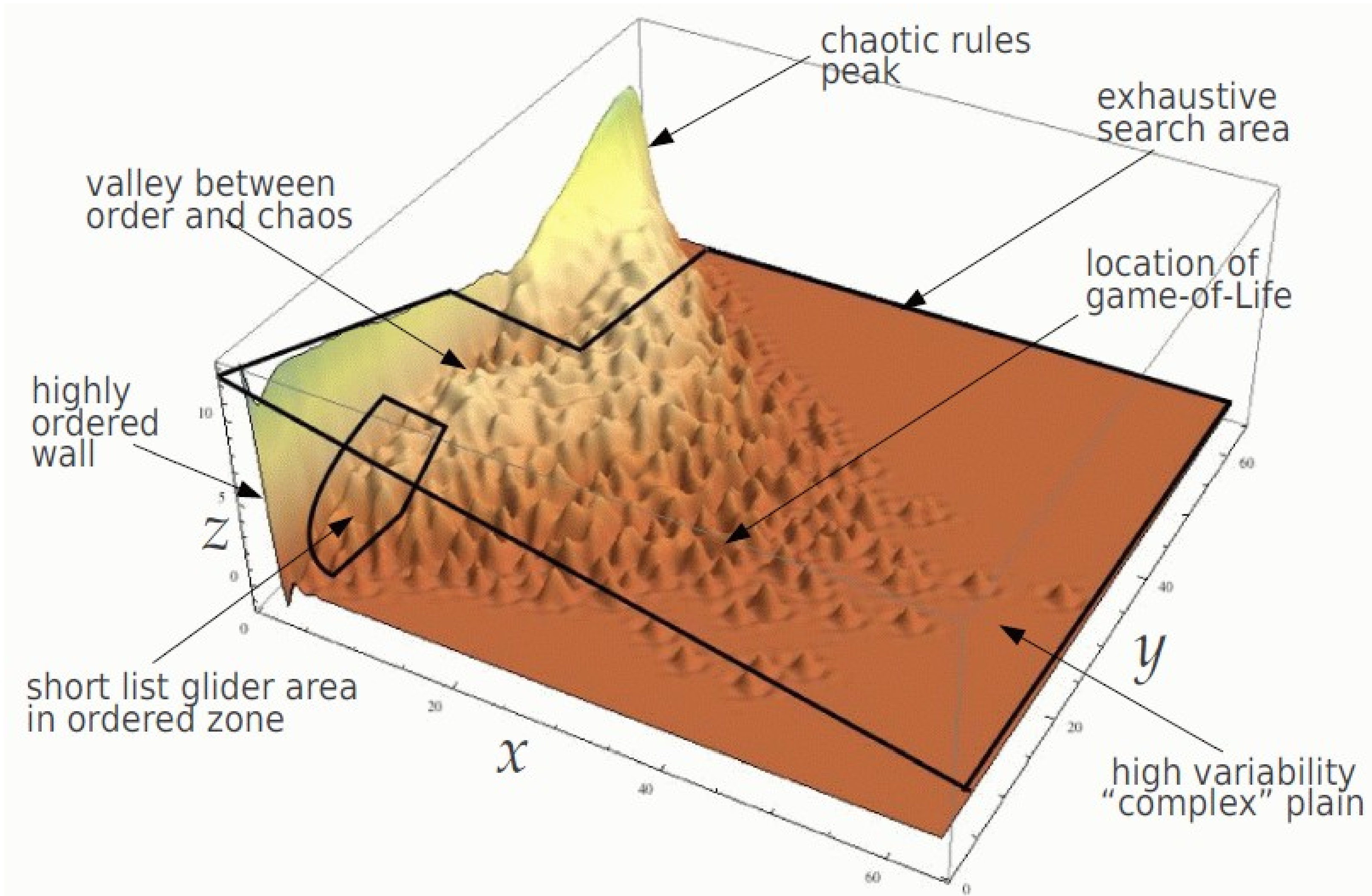
input-entropy and min-max variability

game-of-Life from random 40x40 initial state, density=30%
250 time-steps, colors follow the iso-histogram.
min-max is the biggest upslope ignoring initial (22) steps



automatically classifying iso-rule-space: scatter-plots

find glider/eater iso-rules to construct glider-guns,
(or find emergent glider-guns) for logically universal CA

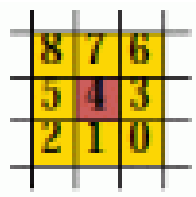


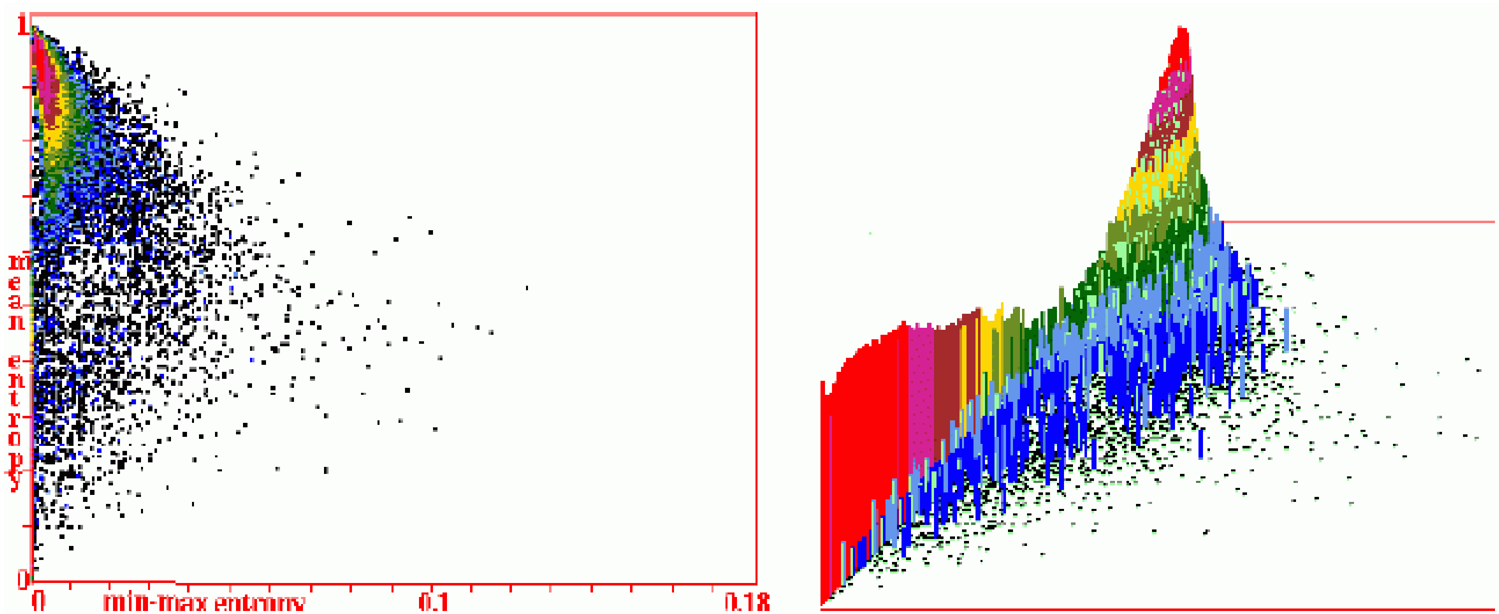
x=entropy-vatiability, y=nean-entropy, z=$\log_2$ rule frequency

iso-rule scatter-plots, lattice 60x60, sample size=50000

sorted by min-max, then by mean entropy

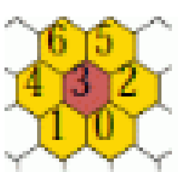
v2k9

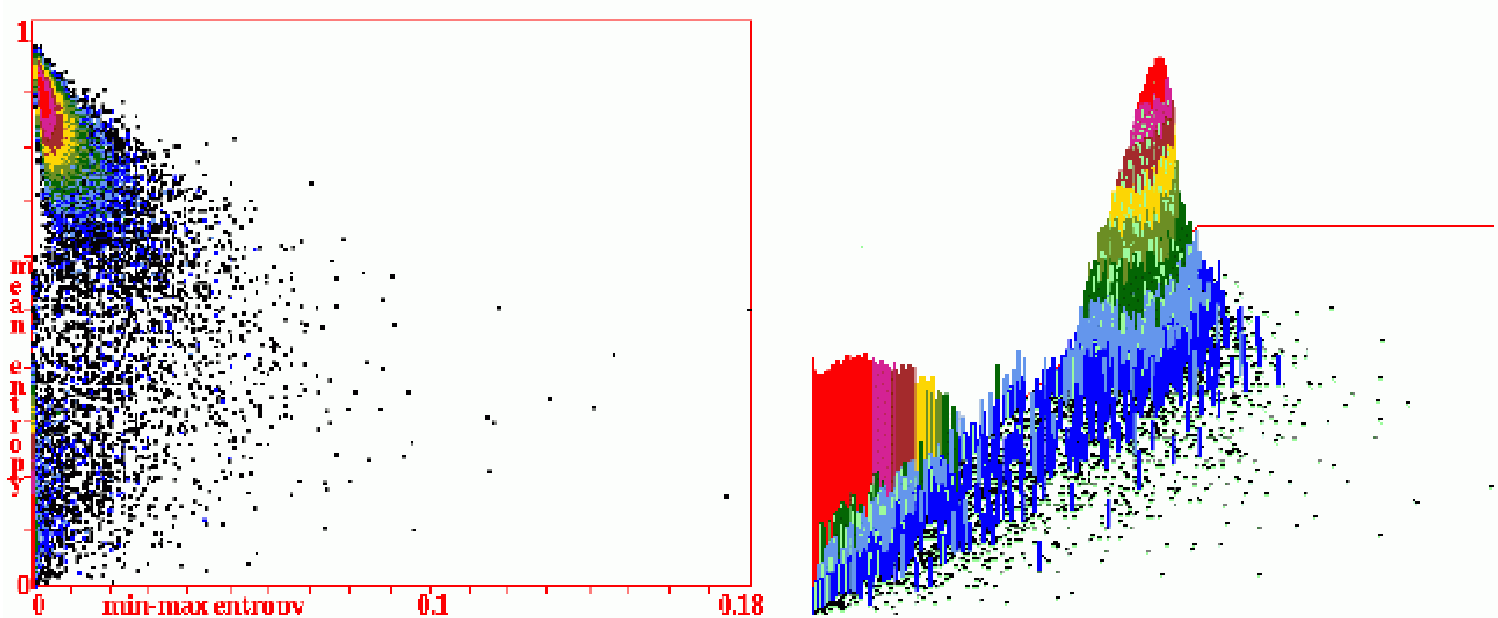




(a) orthogonal lattice $v2k9$, approx density 0/1: seed=70/30, iso-rule=70/30.

v3k7



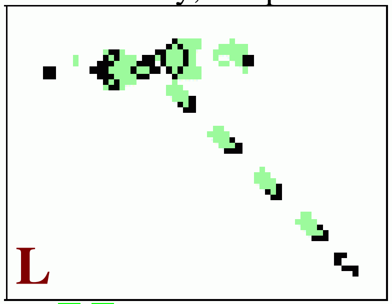


(b) hexagonal lattice $v3k7$, approx density 0/1/2: seed=70/15/15, iso-rule=60,20,20.

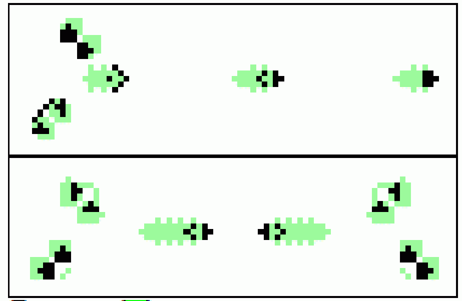
glider-gun review – very few -- Emergent core, Logically universal, *p*=period

v2k9 rules

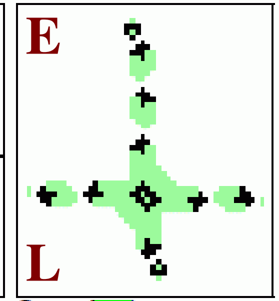
<Conway, Gosper>



Life[3, 7]: p=30



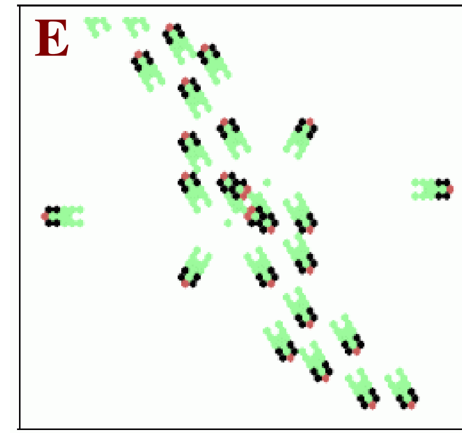
Eppstein[6]: p=68



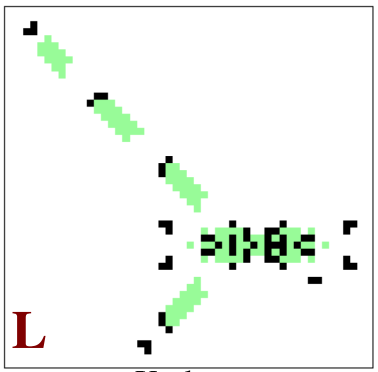
Sapin[17]: p=18

v3k6

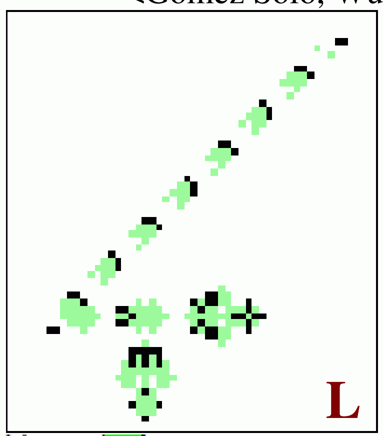
Beehive rule *p*=13 <Wuensche>



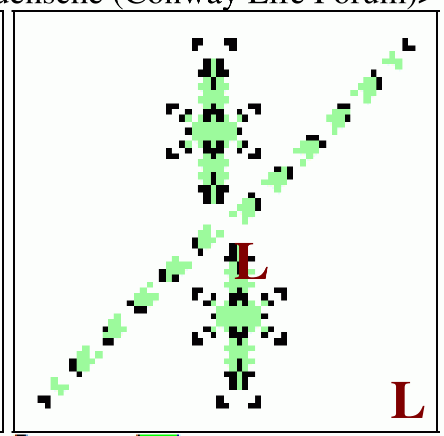
<Gomez Solo, Wuensche (Conway Life Forum)>



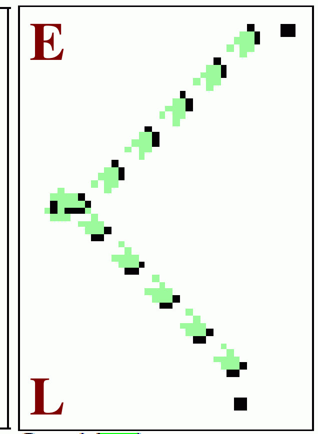
Xrule
not fully isotropic *p*=38



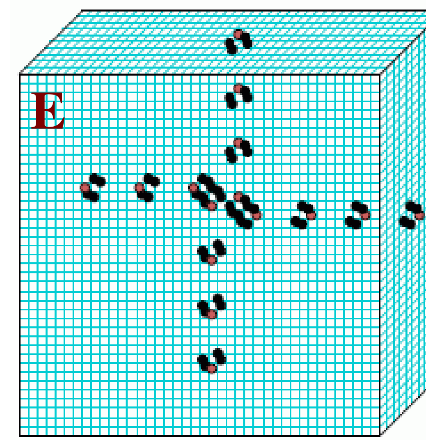
Variant[12]: p=22



Precursor[10]: p=19

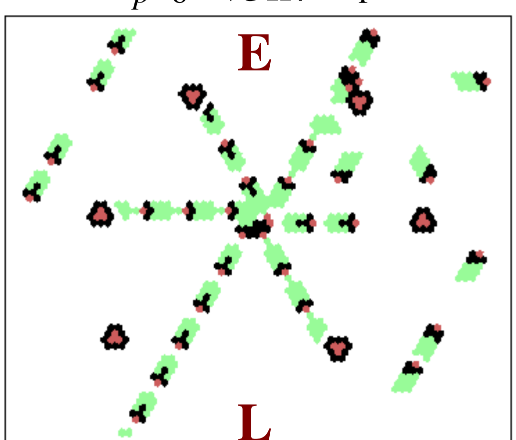


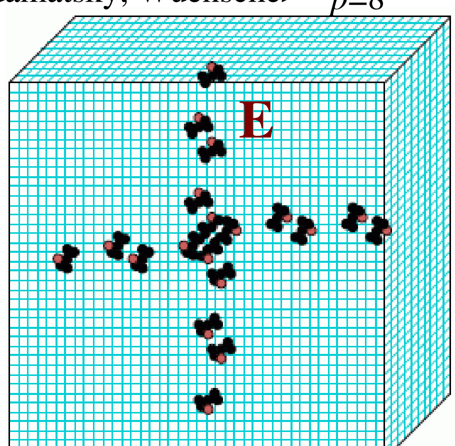
Sayab[11]: p=20



Beehive rule *p*=6 <Wuensche>

p=6 v3k7 Spiral rule <Adamatsky, Wuensche> *p*=8





2 types of glider-guns: constructed, and emergent by glider/object collisions.
Located objects (eaters, reflectors, oscillators) are required to make logical gates.

Discrete Dynamics Lab

OPEN SOURCE Tools for researching Cellular Automata, Random Boolean Networks, multi-value Discrete Dynamical Networks, and beyond

www . ddlab . org

<u>Language:</u>

plain C

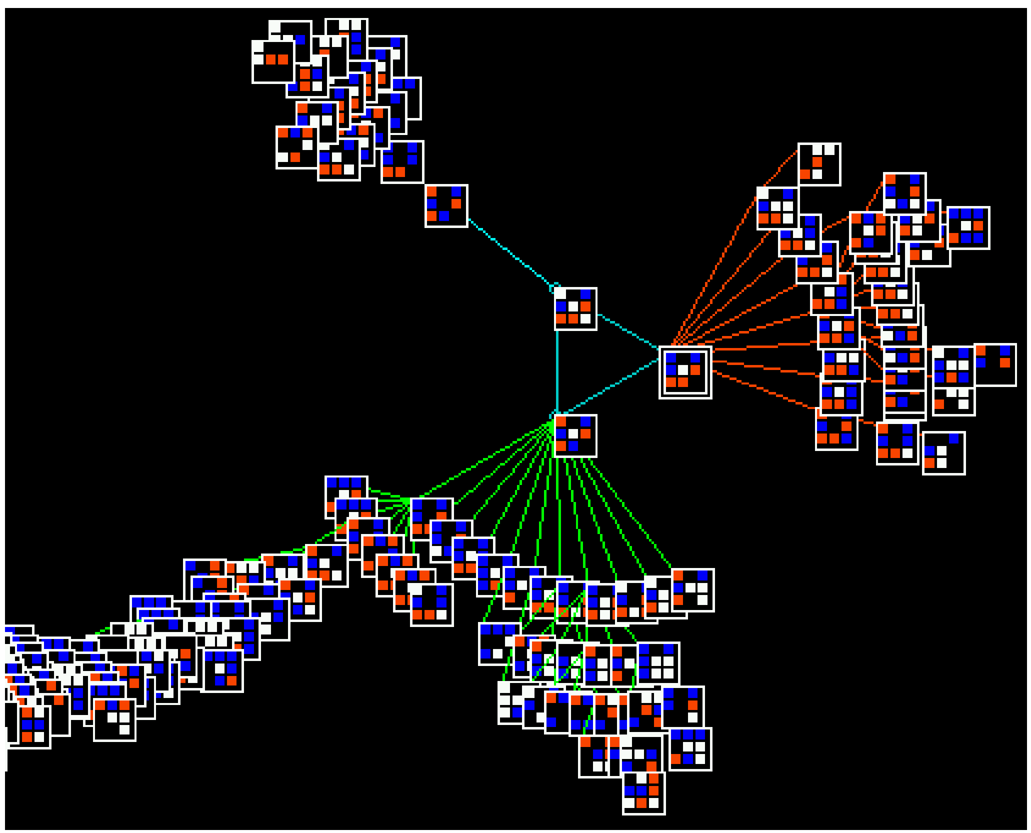
<u>Platforms</u>:

Linux

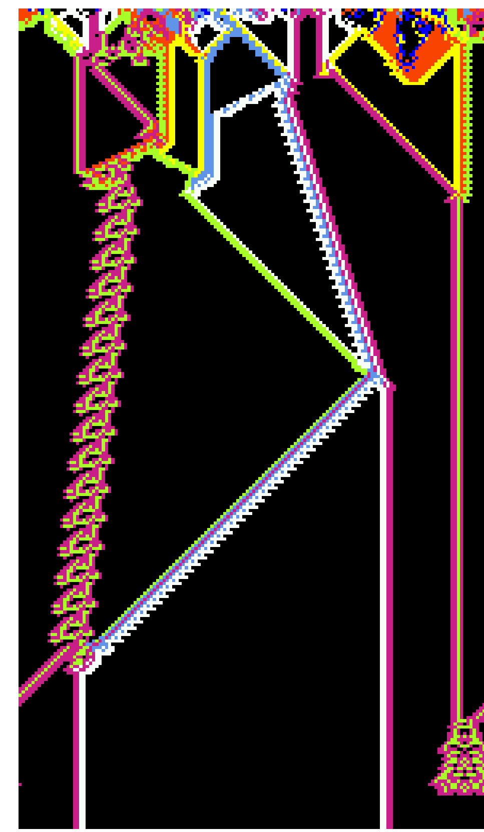
Mac

Cygwin

Dos



basins of attraction



space-time patterns