Flexible constrained sampling with guarantees for pattern mining

Vladimir Dzyuba, DTAI, KU Leuven
Matthijs van Leeuwen, LIACS, Leiden University
Luc De Raedt, DTAI, KU Leuven

ECMLPKDD'17 Journal Track

September 19, 2017

We present *Flexics*

The first pattern sampler that

- supports arbitrary constraints
- supports arbitrary sampling distributions
- provides guarantees on accuracy and efficiency
- goes beyond sampling of individual patterns: it allows to sample k-pattern sets

We regard pattern mining as a tool for exploratory data analysis

Patterns are interpretable descriptions of interesting data regions

What is *interesting*?

Formal problem statement

Given

- $\mathcal{D}$ Dataset
- $\mathcal{C}$ Constraints on patterns (*length, closedness*...)
- φ Measure of pattern quality (*frequency, discriminativity*...)

Find

Patterns that satisfy all constraints in $\mathcal{C}$
and have high quality as measured by φ, e.g., top-k

Properties of a good…

Result set

Compact	Easy to inspect for a human user
Interesting	Satisfies $\mathcal{C}$ and 'high' φ

Mining algorithm

Efficient	User's patience is limited

Properties of a good...

Result set

Compact	Easy to inspect for a human user
Interesting?	Satisfies $\mathcal{C}$ and 'high' φ

Mining algorithm

Efficient	User's patience is limited

Ideal user

I am only interested in top-50 patterns according to purity that are closed and have frequency above 40 and length above 7

Real-world user

Well, they should be frequent enough, maybe 40, and probably have more $\oplus$s than $\ominus$s... Purity? Closed? Can I just see a few first?

Properties of a good...

Result set

Compact	Easy to inspect for a human user
Interesting	Satisfies $\mathcal{C}$ and 'high' φ
Diverse	Allows for serendipity and avoids redundancy

Mining algorithm

Efficient	User's patience is limited
Flexible	Supports exploring various constraints and quality measures

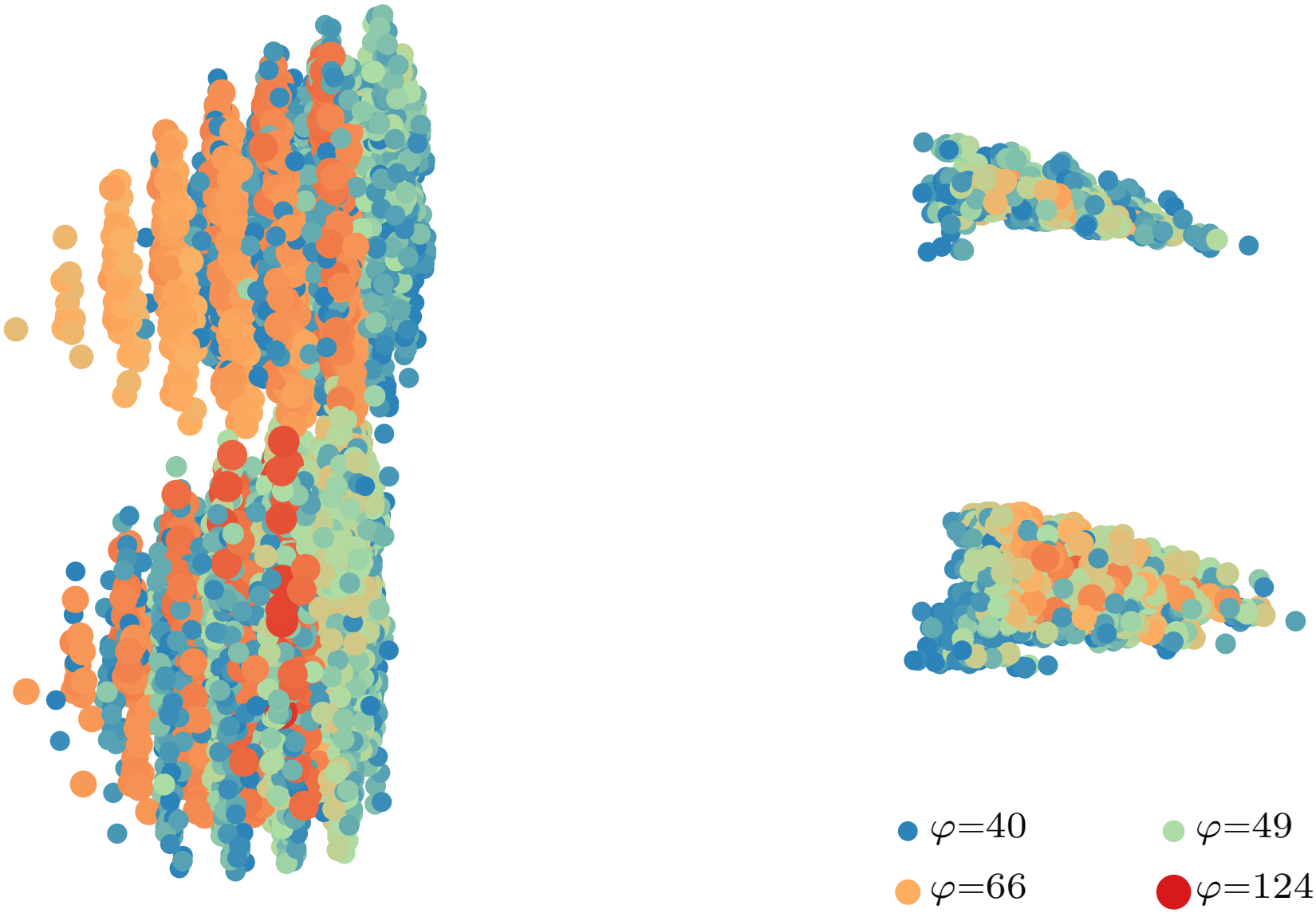
Running example from itemset mining

$$\mathcal{D} = \texttt{vote}$$

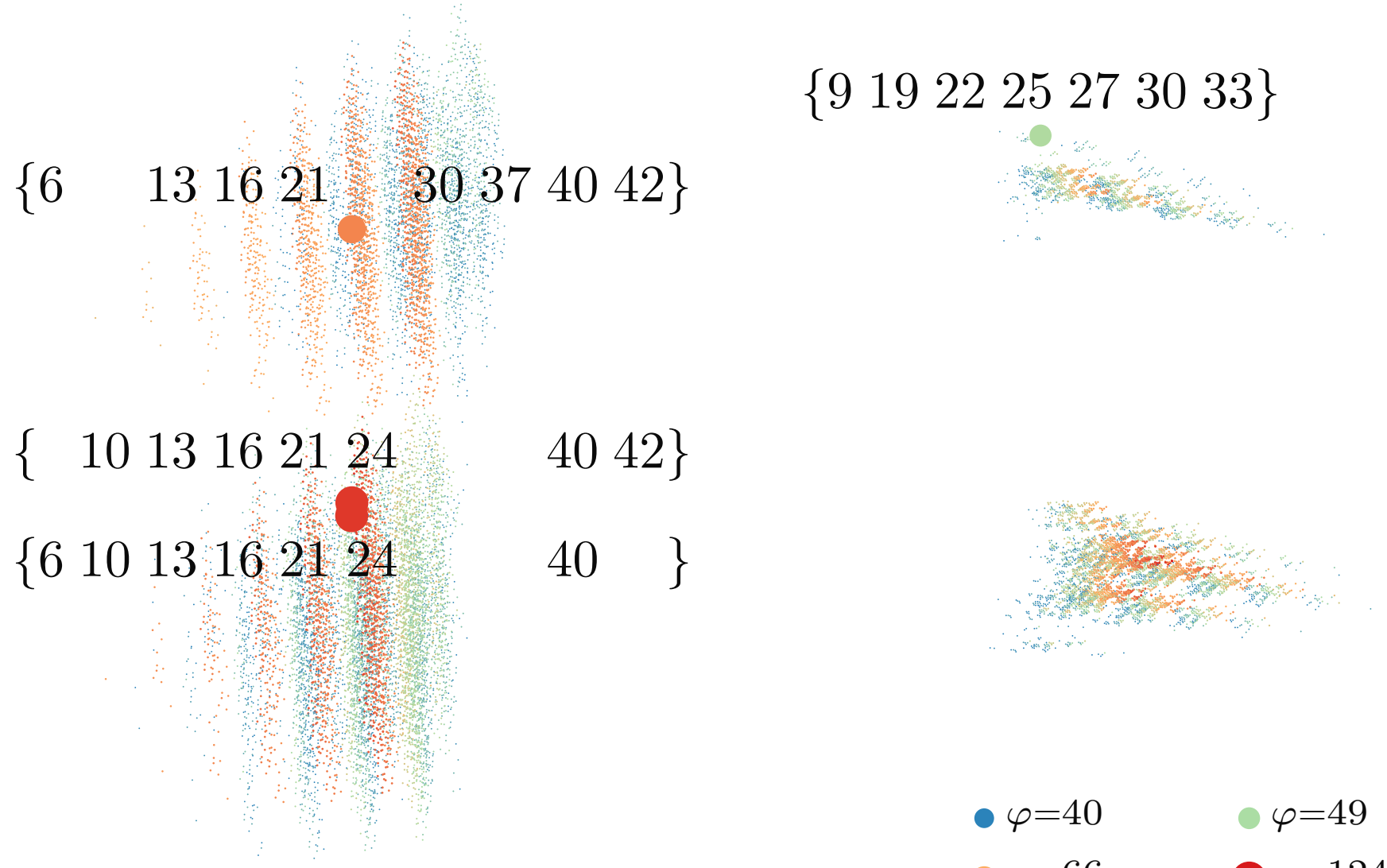
$$\mathcal{C} = \{Frequency \geq 40, \\ Closed, Length \geq 7\}$$

$$\varphi\left(p\right) = Frequency\left(p\right)$$

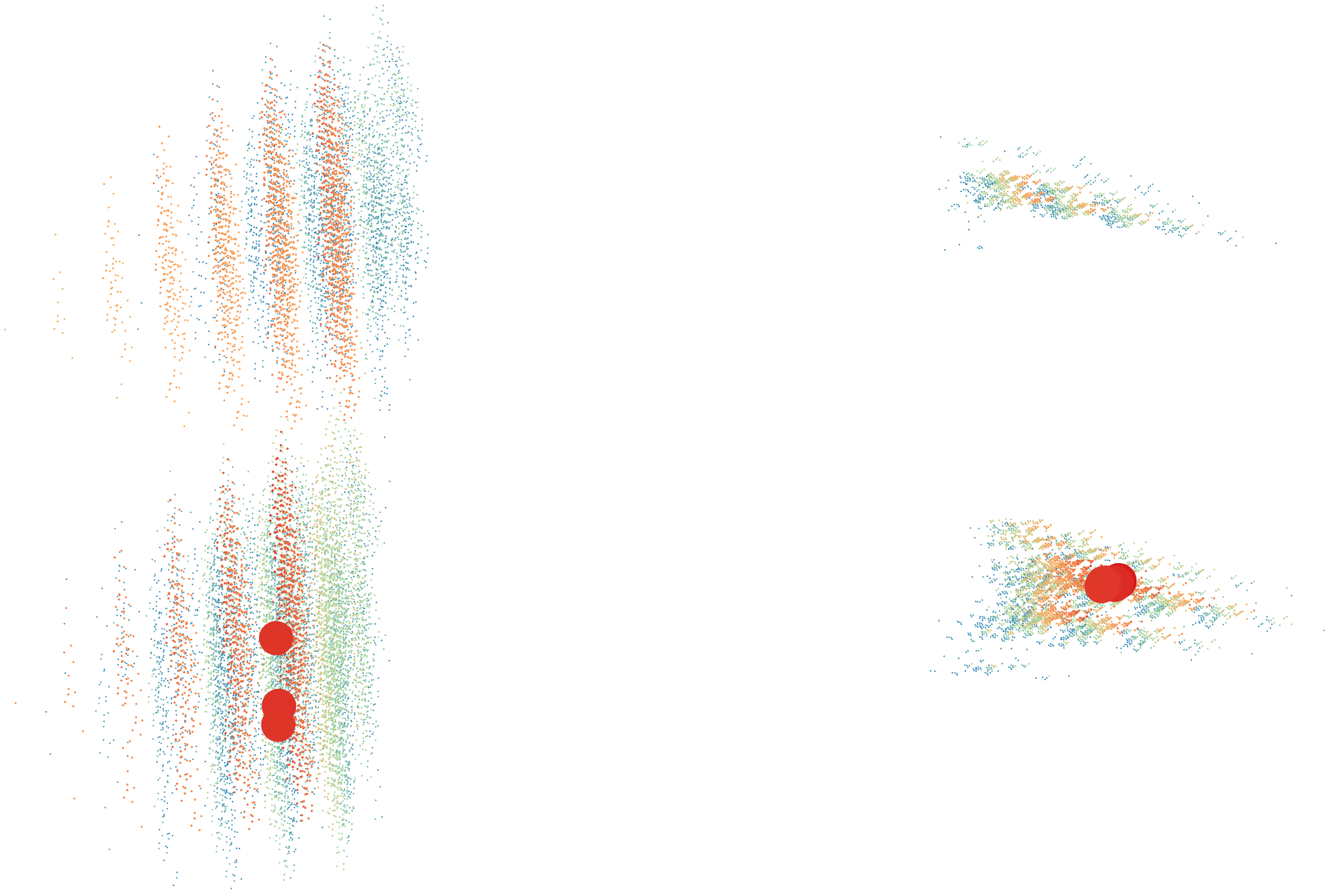
Example: pattern space illustration



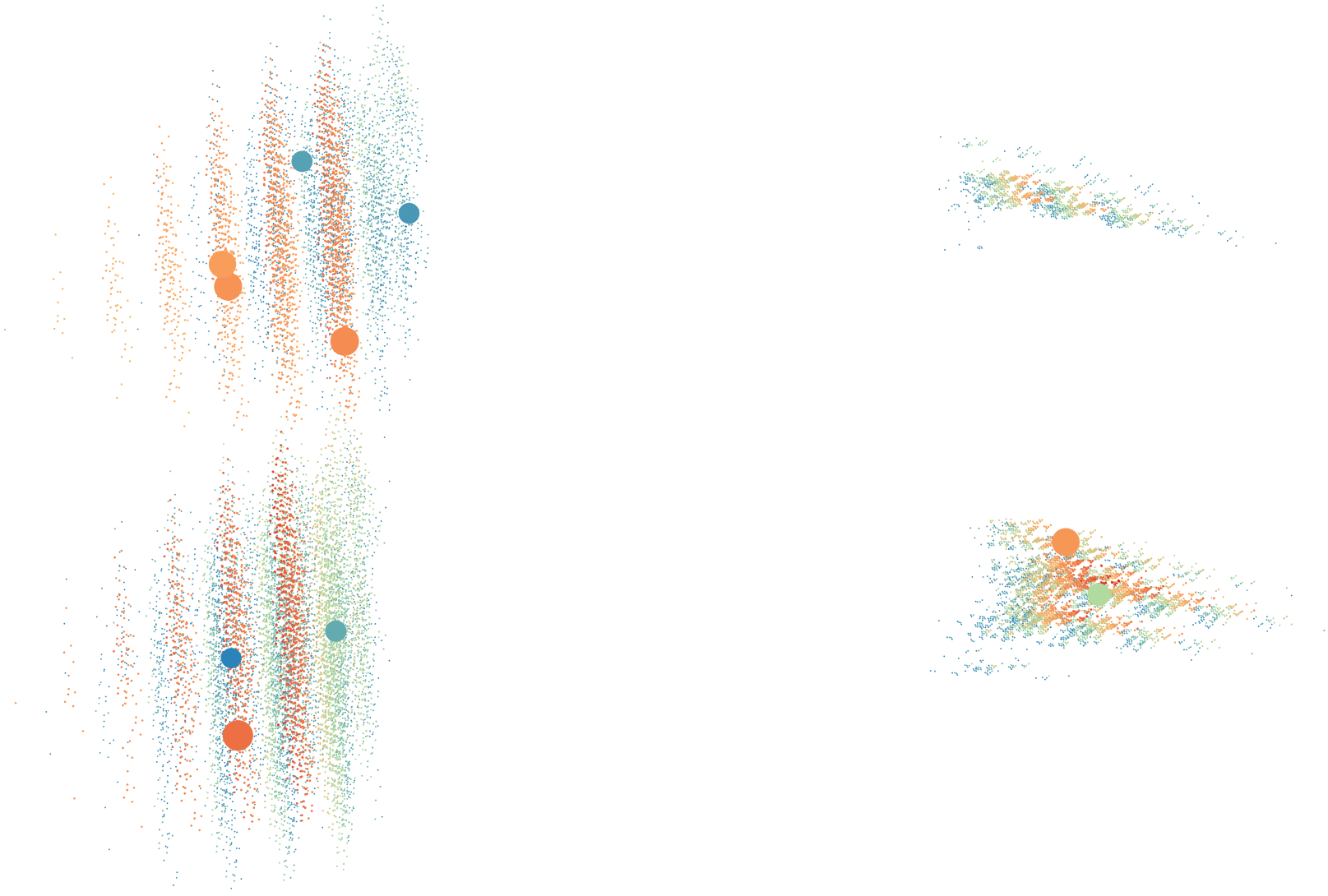
Example: pattern space illustration



Top-10 is not diverse/representative



More representative set of 10 patterns



Pattern sampling

Given

- $\mathcal{D}$ Dataset
- $\mathcal{C}$ Constraints on patterns (*length*, *closedness*...)
- φ Measure of pattern quality (*frequency*, *discriminativity*...)

Generate randomly

Patterns satisfying $\mathcal{C}$ proportional to their quality φ:

$$P(p \text{ is in the result set}) \propto \varphi(p)$$

Benefits of pattern sampling

Result set is…

Compact	Growing result set by request
Interesting	Respects $\mathcal{C}$ and φ
Diverse	"Naturally", owing to randomization

Sampling is…

Efficient	'Anytime': sample patterns one by one

Benefits & challenges of pattern sampling

Result set is…

Compact	Growing result set by request
Interesting	Respects $\mathcal{C}$ and φ
Diverse	"Naturally", owing to randomization

Sampling is…

Efficient	'Anytime': sample patterns one by one
Flexible	How to sample given *arbitrary* $\mathcal{C}$ and φ?

Flexics

Flexible **c**onstrained pattern **s**ampler

- Supports black-box quality measures...
- ...in combination with arbitrary constraints
- Provides accuracy and runtime guarantees

High-level sampling procedure based on hashing
(Chakraborty et al. 2014)

1. Partition the pattern space into random non-overlapping "*cells*"
2. Enumerate patterns in a randomly chosen cell
3. Return a perfect sample from the cell

How to obtain "good" random cells?

Append *random* XOR constraints on pattern descriptions to $\mathcal{C}$

k XOR constraints $\Leftrightarrow$ one out of 2^k "cells"

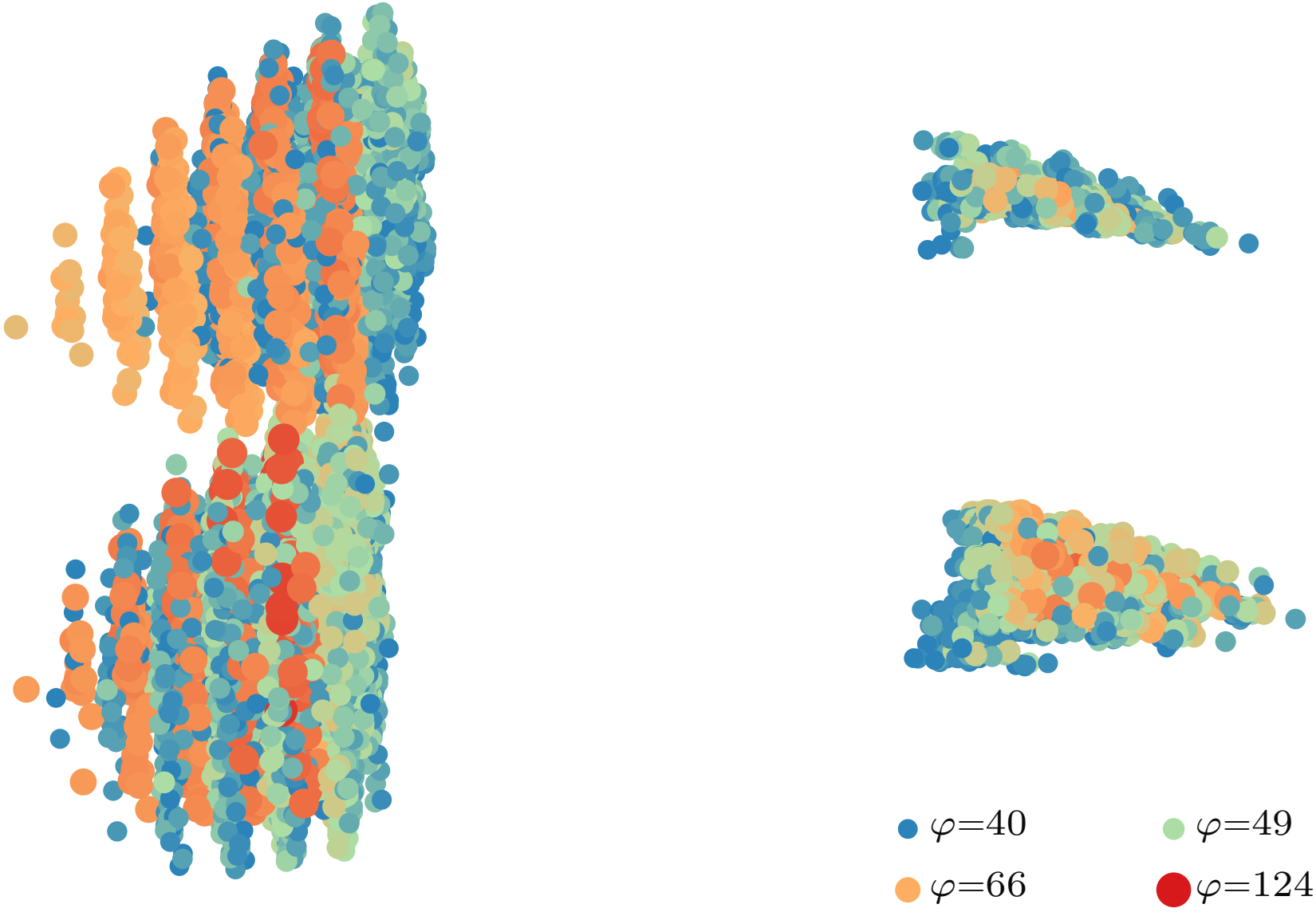
$$\begin{cases} \bigotimes \underbrace{b_{1i} \cdot p_i}_{0/1} = b_{10} \\ \bigotimes b_{2i} \cdot p_i = b_{20} \\ \dots \\ \bigotimes b_{ki} \cdot p_i = b_{k0} \end{cases}$$

$p_i = 1$ if item $i \in$ pattern p

Flipping a coin for coefficients $b_{0|i}$ yields "good" random cells

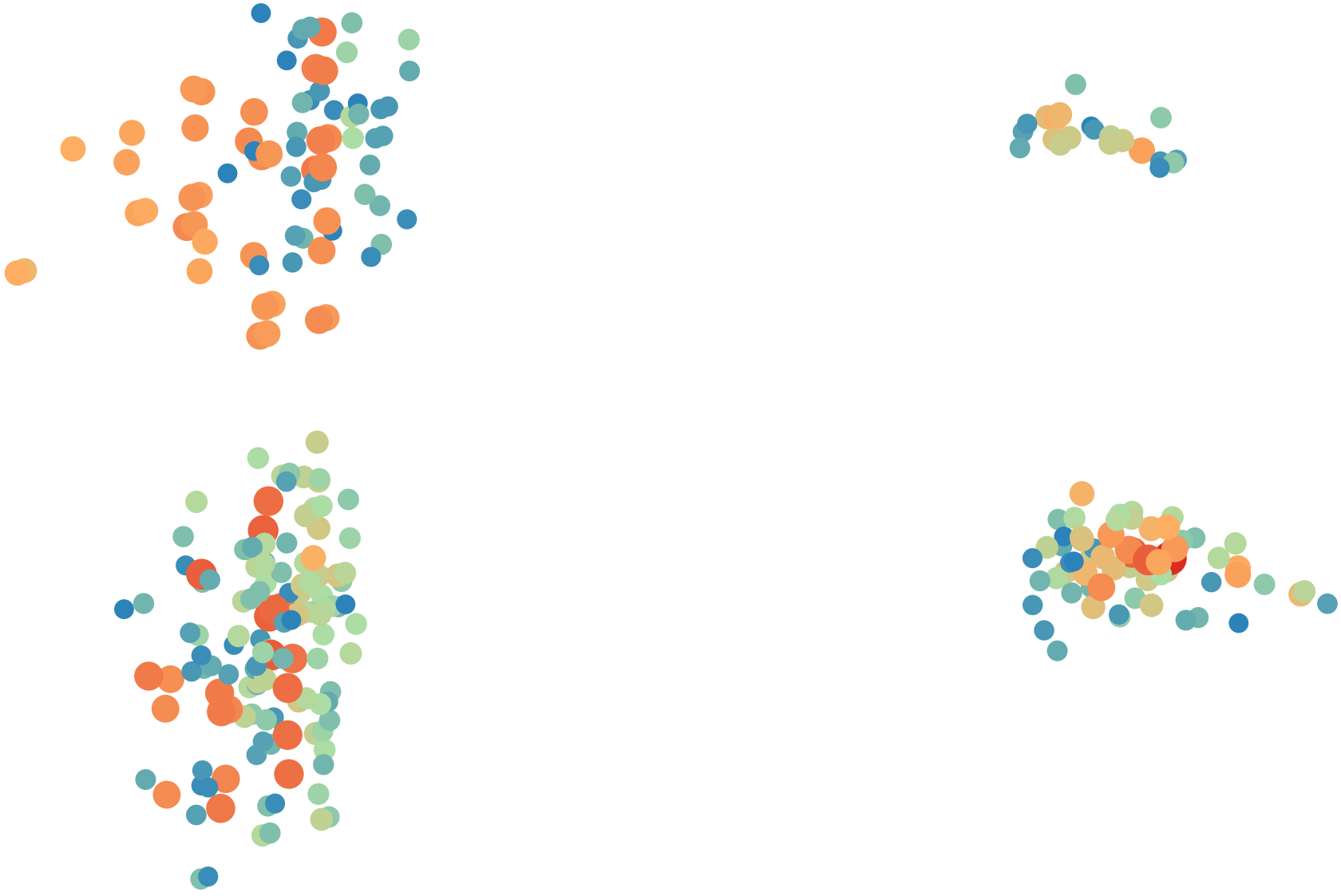
Cell membership is independent of pattern description

$N_{XOR} = 6$ determined during pre-processing



Cell membership is independent of pattern description

$N_{XOR} = 6$ determined during pre-processing



Many miners can be augmented with XOR constraint handling and used as a "backend" for *Flexics*

Uses binary Gaussian elimination

The "backend" doesn't need any knowledge about φ!

Two "backends" = two variants of Flexics

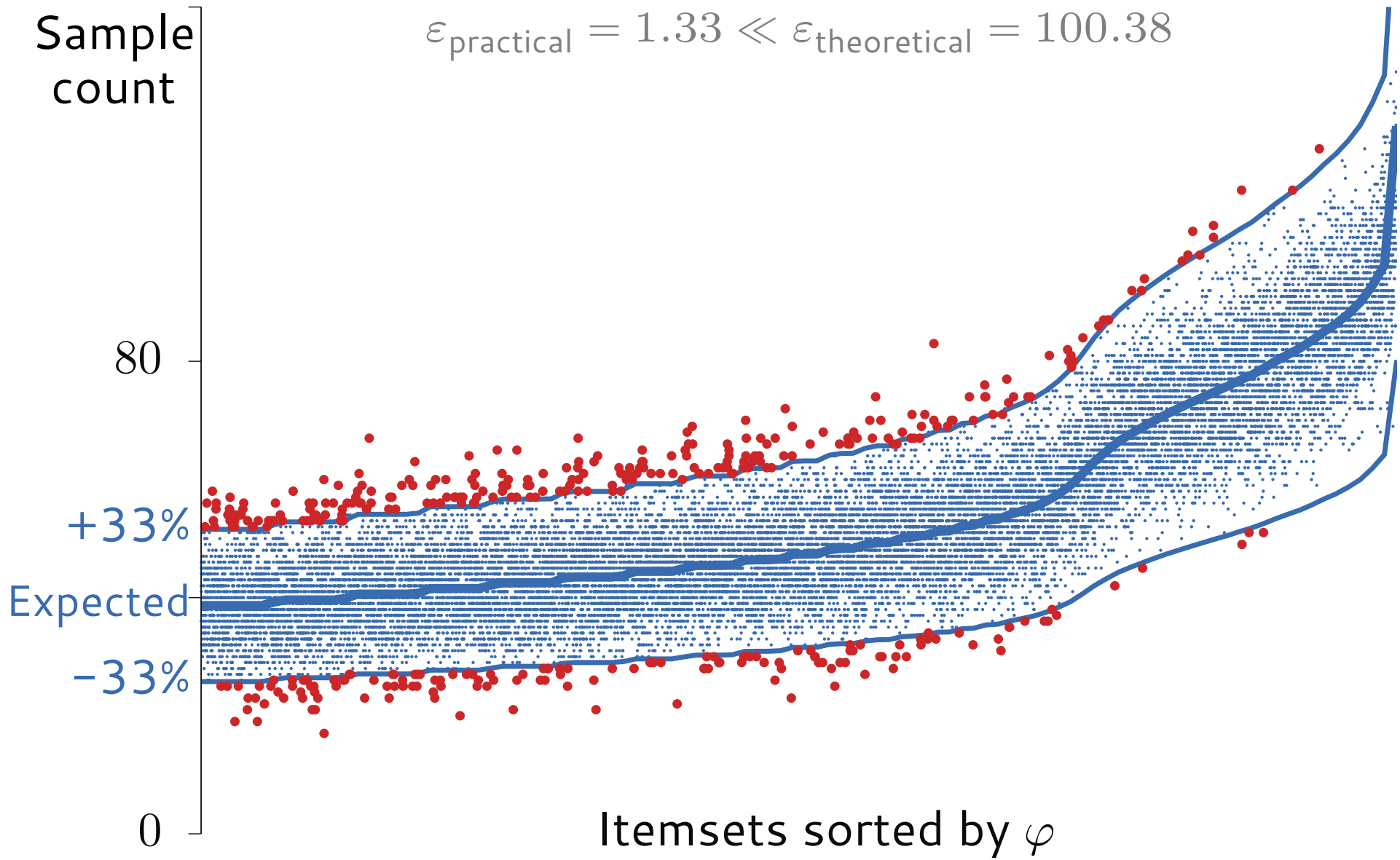
- CP4IM (Guns et al. 2011) $\Rightarrow$ *GFlexics*
  Generic mining system based on *constraint programming*
  Supports a wide range of **constraints & pattern types**: itemsets, tilings...

- Eclat (Zaki et al. 1997) $\Rightarrow$ *EFlexics*
  Efficient frequent itemset sampler

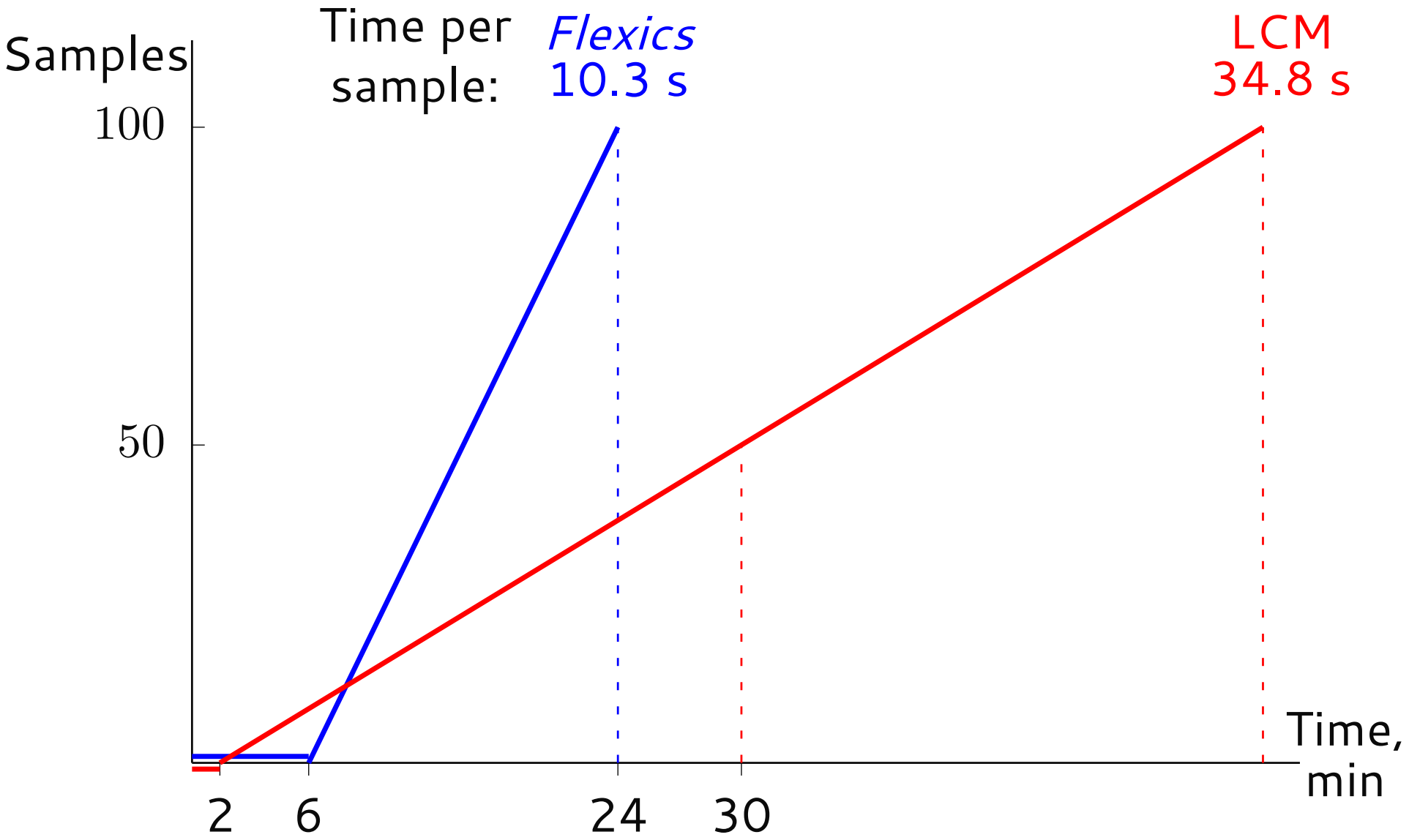
Flexics provides
theoretical guarantees
on both sampling error and the
number of backend calls
(with any backend)

Experimental evaluation

Flexics samples with high accuracy

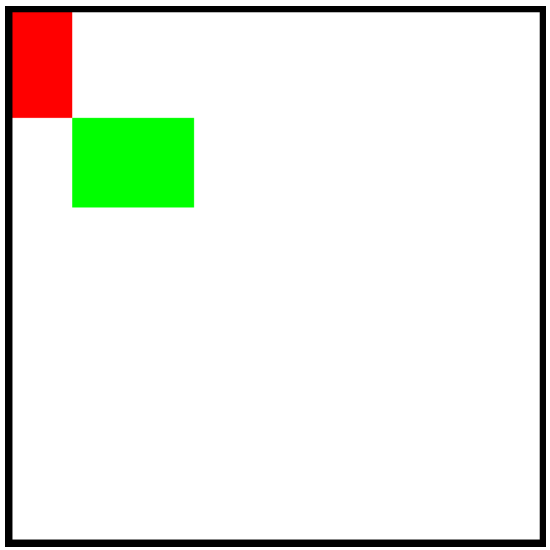


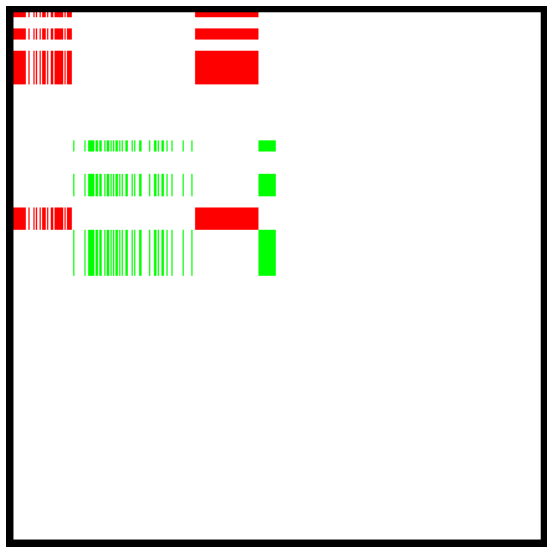
Flexics can handle moderately large data

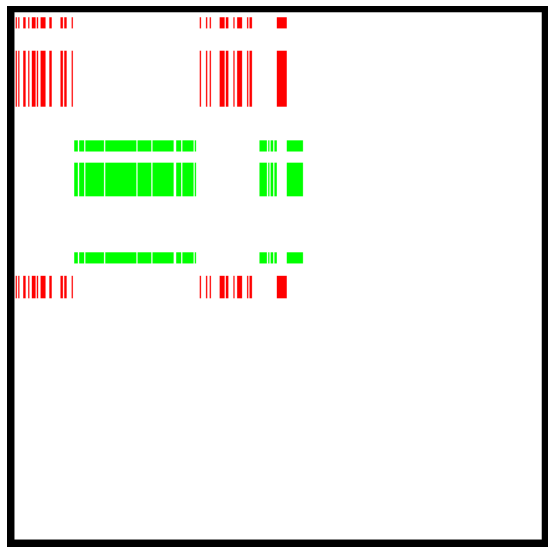


Flexics supports sampling of complex patterns

Sampling 2-tilings proportional to their area







Conclusions

- *Flexics* is efficient and flexible,

  which compensates for not knowing upfront what is interesting in both supervised and unsupervised data exploration, and

  makes pattern mining practically more applicable

- Simple procedure enables further tinkering,
  e.g., in interactive mining (Dzyuba & van Leeuwen 2017)

- Future work includes reducing effect of *tilt*
  and extending to richer pattern languages, e.g., sequences