Supervised Pattern Set Mining

Albrecht Zimmermann
DTAI (ML)
Katholieke Universiteit Leuven

albrecht.zimmermann@cs.kuleuven

Overview

Differences/Similarities to Unsupervised PSM

- Supervised measures

Partitioning the data

Four ways of going about it

- CBA
- DTM
- fCork
- ReMine

Related issues

What's different?

Different data sets

- i.e. different points in time, locations

⇔

Data split into subsets

- different classes
- subgroups w.r.t. target attribute

We discuss only the binary case

Tasks related to target

- find contrasting patterns
- class prediction
- describe subgroups for further offline analysis

Unsupervised methods may be able to help as well

(Pseudo-)Notation

Not only itemsets

- change matching-relation

$$X \subseteq t \rightarrow P \preceq t$$

We don't care about pattern language

$$cov(P) = \{t \mid P \preceq t\}$$

"Naming" the subsets

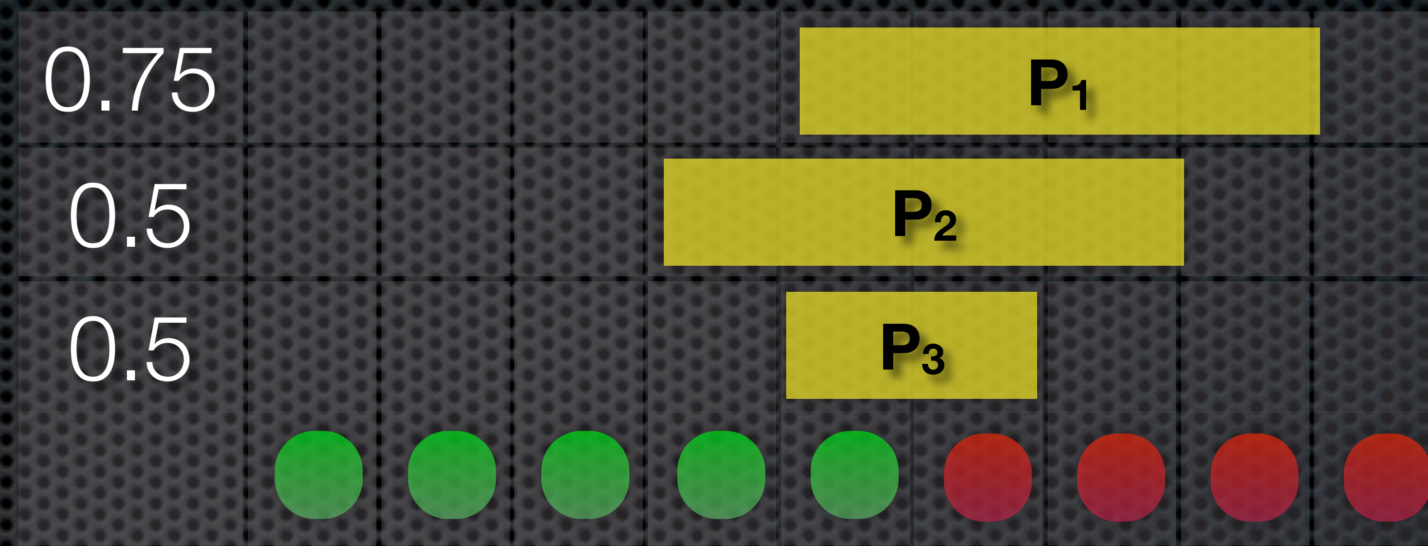
$$class_1 = db \cap \{\ \}$$

$$class_2 = db \cap \{\ \}$$

Supervised Measures

Accuracy, Confidence

- Conditional probability



$$acc(P) = \frac{\max\{sup_{class_1}(P), sup_{class_2}(P)\}}{sup_{db}(P)}$$

No consideration of coverage

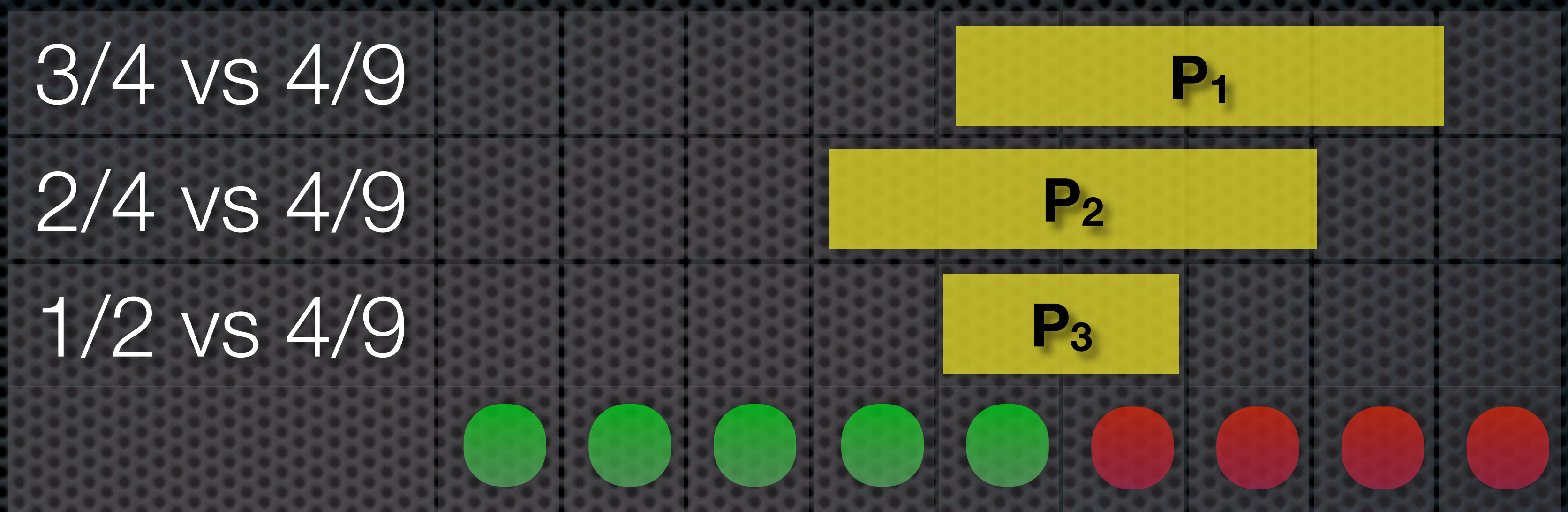
- Augmentation needed
- Upper bound **1**

Supervised Measures

0.991

Correlation

- compare conditional probability to overall probability



Information Gain

- related to entropy

$$\boxed{ent_{class}(db)} = -\sum_{i=1}^{2} \frac{|class_i|}{|db|} \log \frac{|class_i|}{|db|}$$

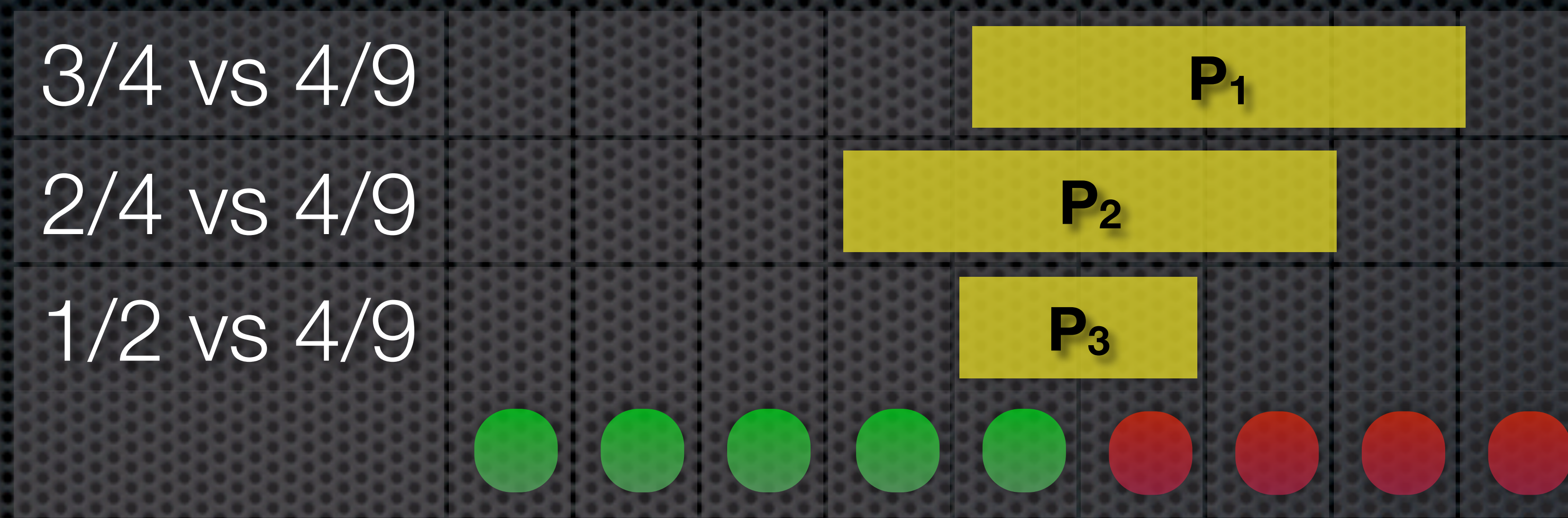
$$IG(P) = ent_c(db) - \frac{|cov(P)|}{|db|} ent_c(cov(P)) - \frac{|db \setminus cov(P)|}{|db|} ent_c(db \setminus cov(P))$$

Supervised Measures

0.991

Correlation

- compare conditional probability to overall probability



Information Gain

- related to entropy

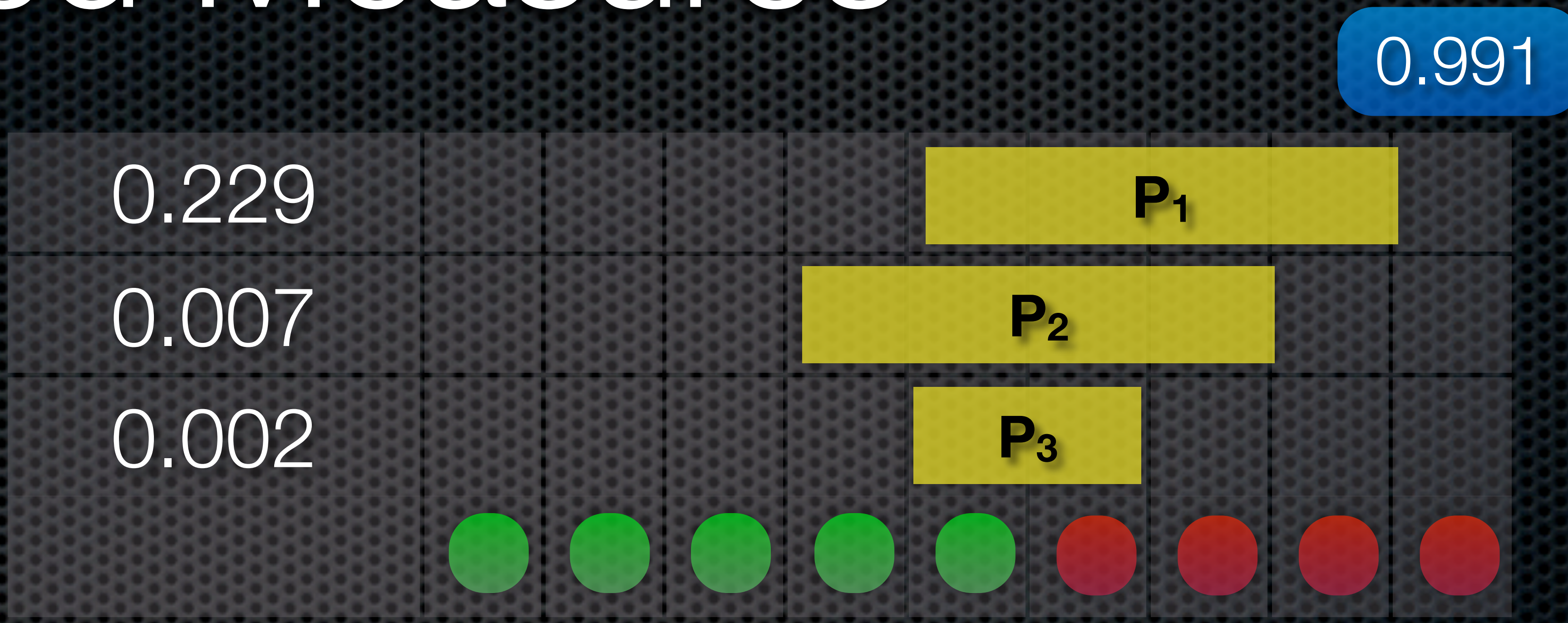
$$ent_{class}(db) = -\sum_{i=1}^{2} \frac{|class_i|}{|db|} \log \frac{|class_i|}{|db|}$$

$$IG(P) = ent_c(db) - \boxed{\frac{|cov(P)|}{|db|}} ent_c(cov(P)) - \boxed{\frac{|db \setminus cov(P)|}{|db|}} ent_c(db \setminus cov(P))$$

Supervised Measures

Correlation

- compare conditional probability to overall probability



Information Gain

- related to entropy

$$ent_{class}(db) = -\sum_{i=1}^{2} \frac{|class_i|}{|db|} \log \frac{|class_i|}{|db|}$$

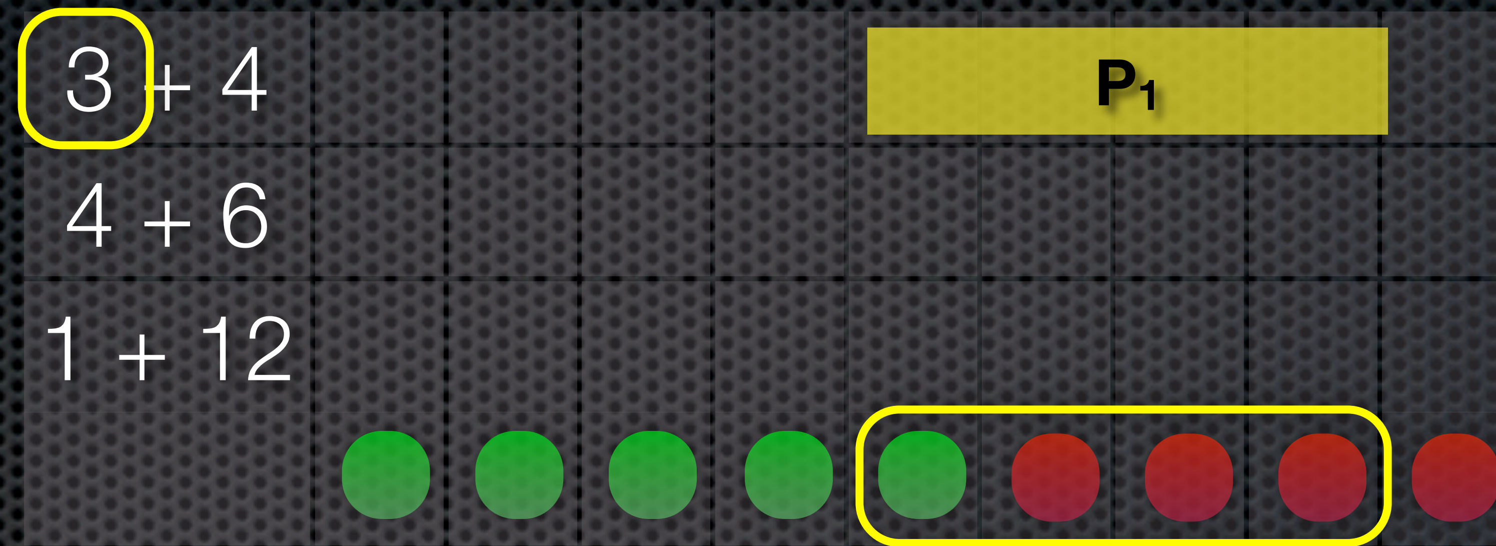
$$IG(P) = ent_c(db) - \boxed{\frac{|cov(P)|}{|db|}} ent_c(cov(P)) - \boxed{\frac{|db \setminus cov(P)|}{|db|}} ent_c(db \setminus cov(P))$$

Upper-boundable

Supervised Measures

Correspondence

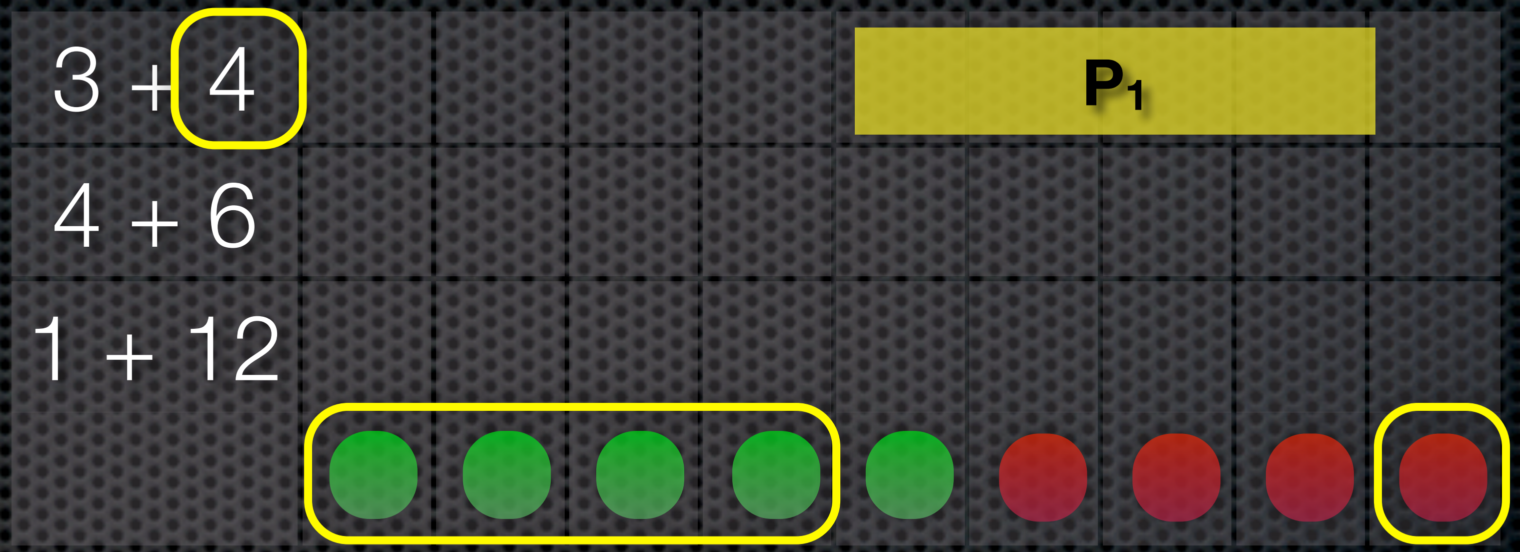
- Count number pairs from different sets



Supervised Measures

Correspondence

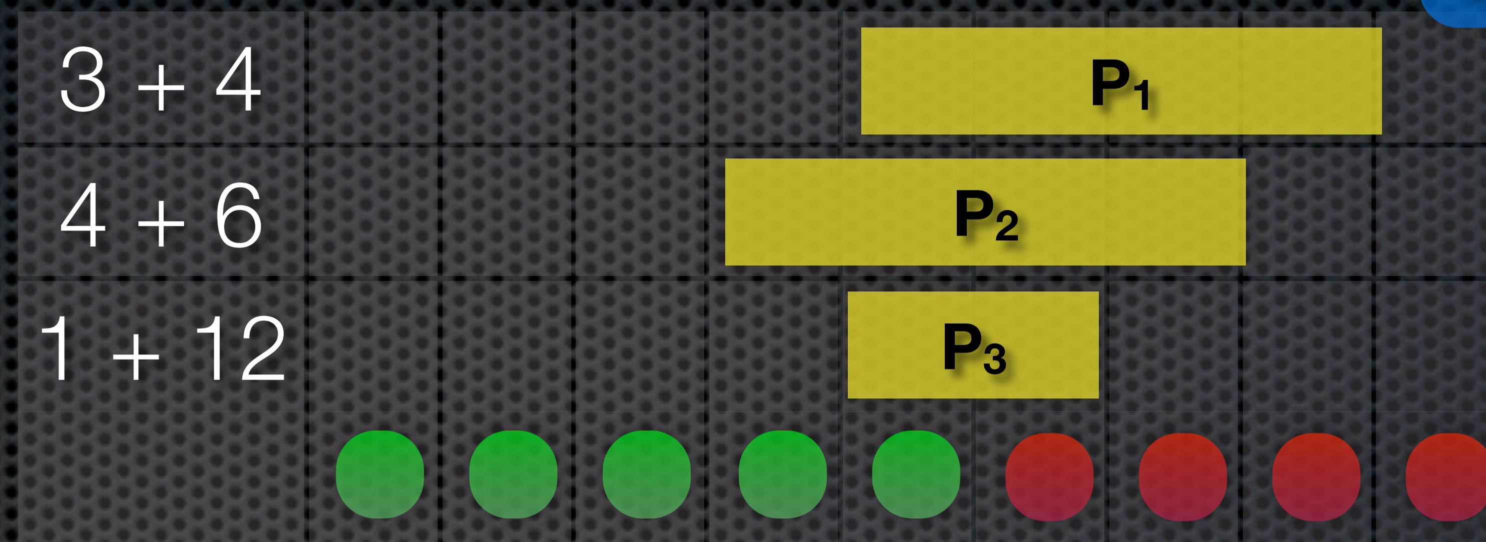
- Count number pairs from different sets



Supervised Measures

Correspondence

- Count number pairs from different sets



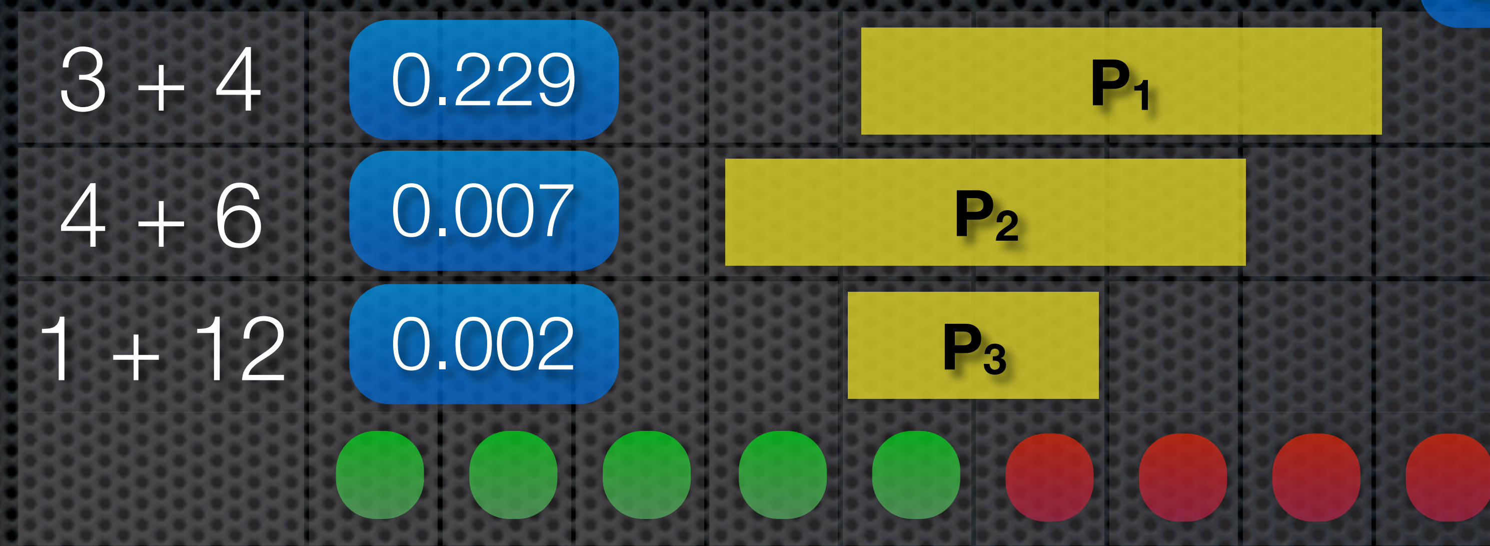
$$corr(P) = sup_{c_1}(P) \cdot sup_{c_2}(P) + (|c_1| - sup_{c_1}(P)) \cdot (|c_2| - sup_{c_2}(P))$$

Supervised Measures

20

Correspondence

- Count number pairs from different sets



$$corr(P) = sup_{c_1}(P) \cdot sup_{c_2}(P) + (|c_1| - sup_{c_1}(P)) \cdot (|c_2| - sup_{c_2}(P))$$

Upper-boundable
Sub-modular

What’s the same?

We want few patterns:

- Alleviating the effect of the curse of dimensionality
- Enhancing generalization capability
- Speeding up learning process
- Improving model interpretability (or description analysis)

We want high-quality patterns

- Predictive or typical

We want little redundancy

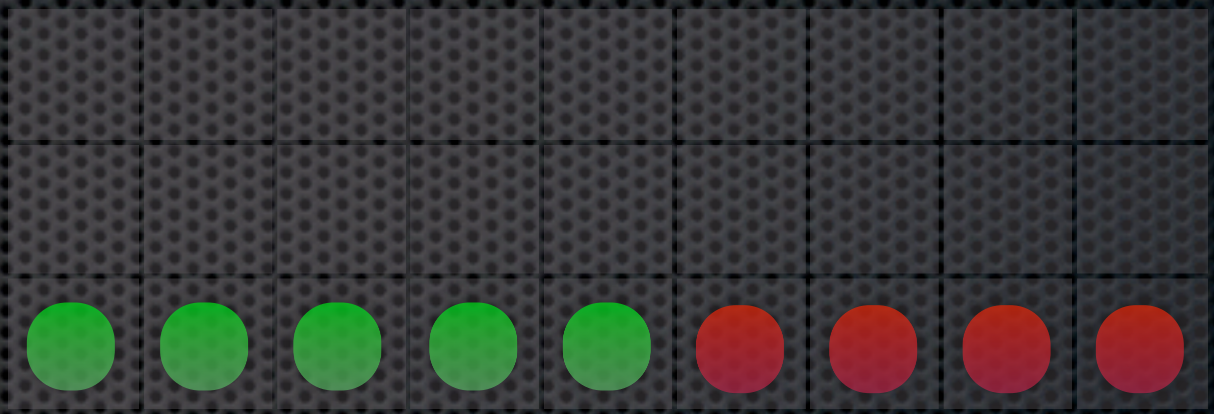
- Discovering same subgroup over and over helps no one

Combination with unsupervised measures possible

Patterns as splits

Imposes new subsets

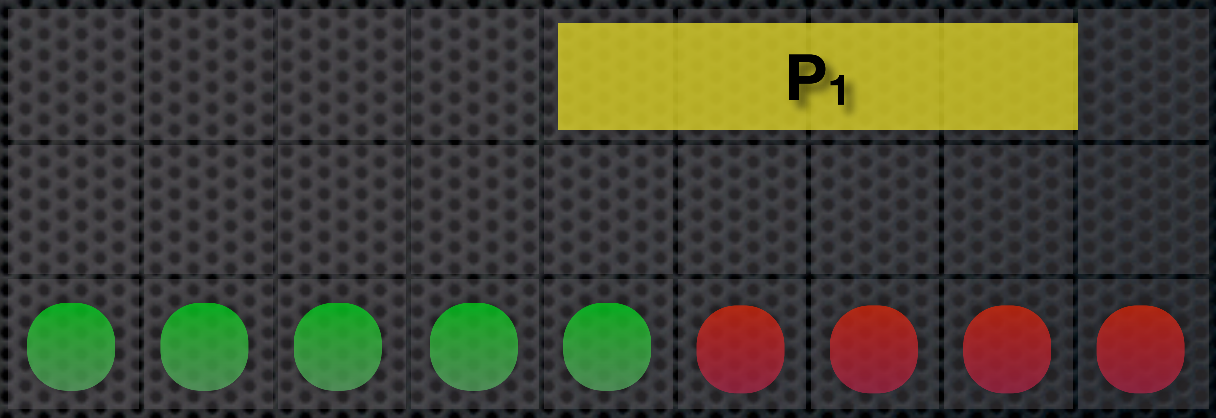
- Pattern absence/ presence becomes new property of data point



Patterns as splits

Imposes new subsets

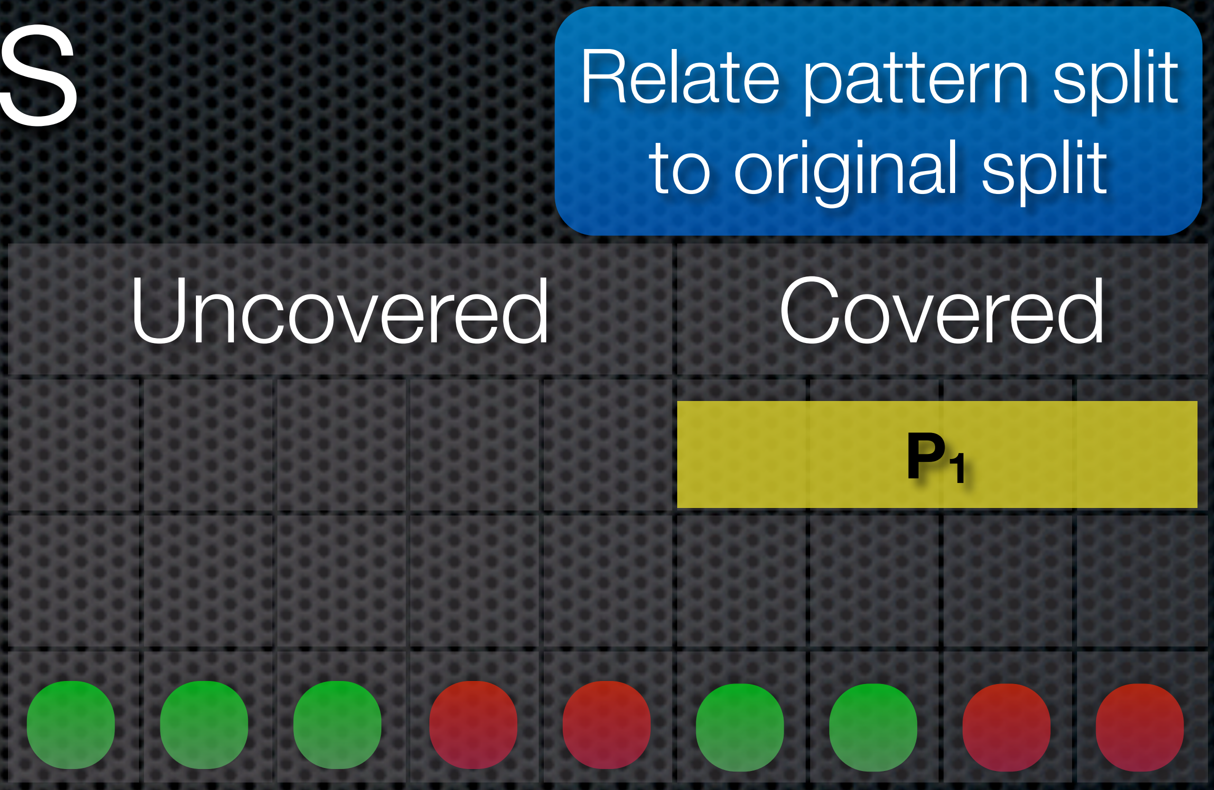
- Pattern absence/presence becomes new property of data point



Patterns as splits

Imposes new subsets

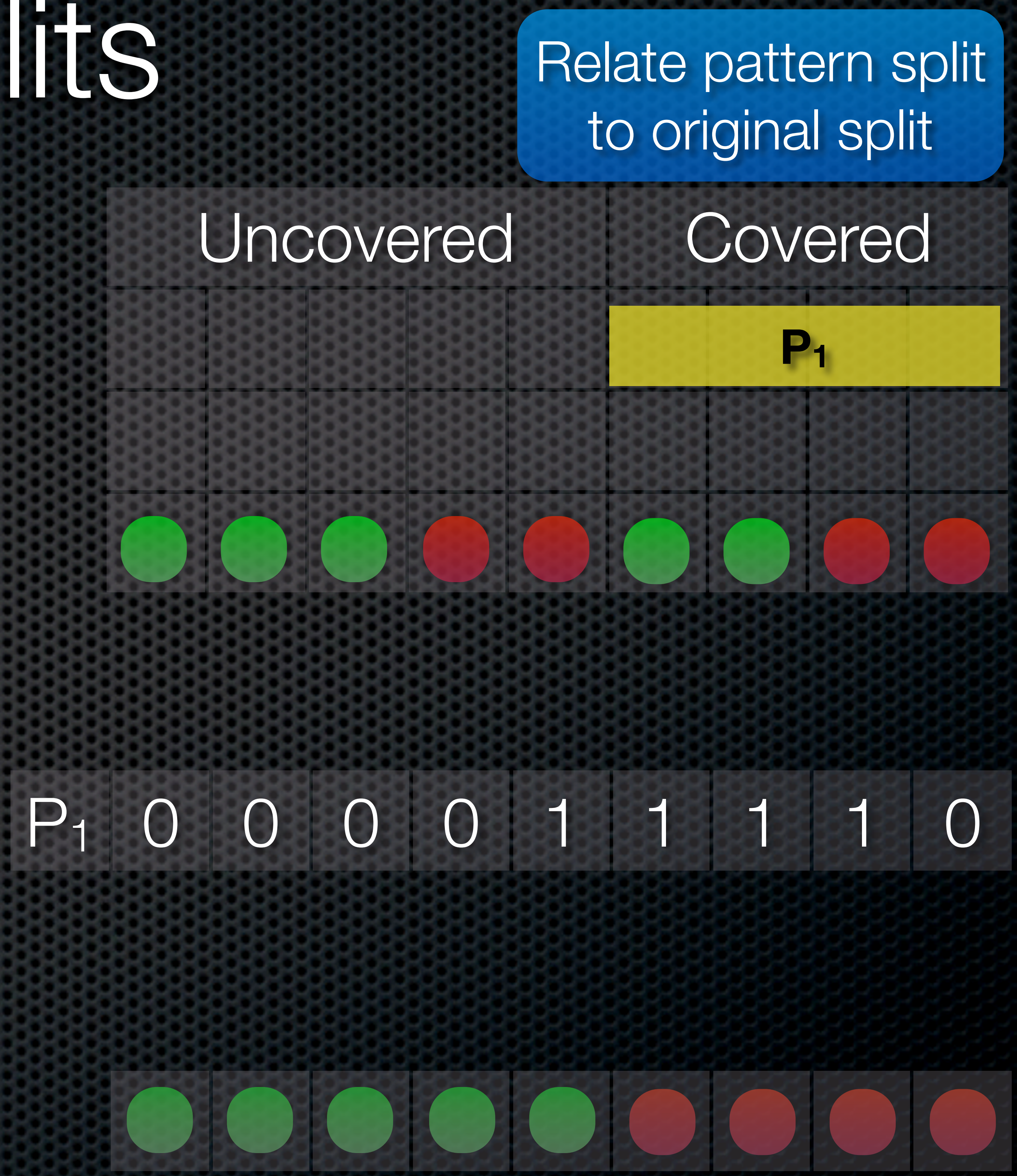
- Pattern absence/ presence becomes new property of data point



Patterns as splits

Imposes new subsets

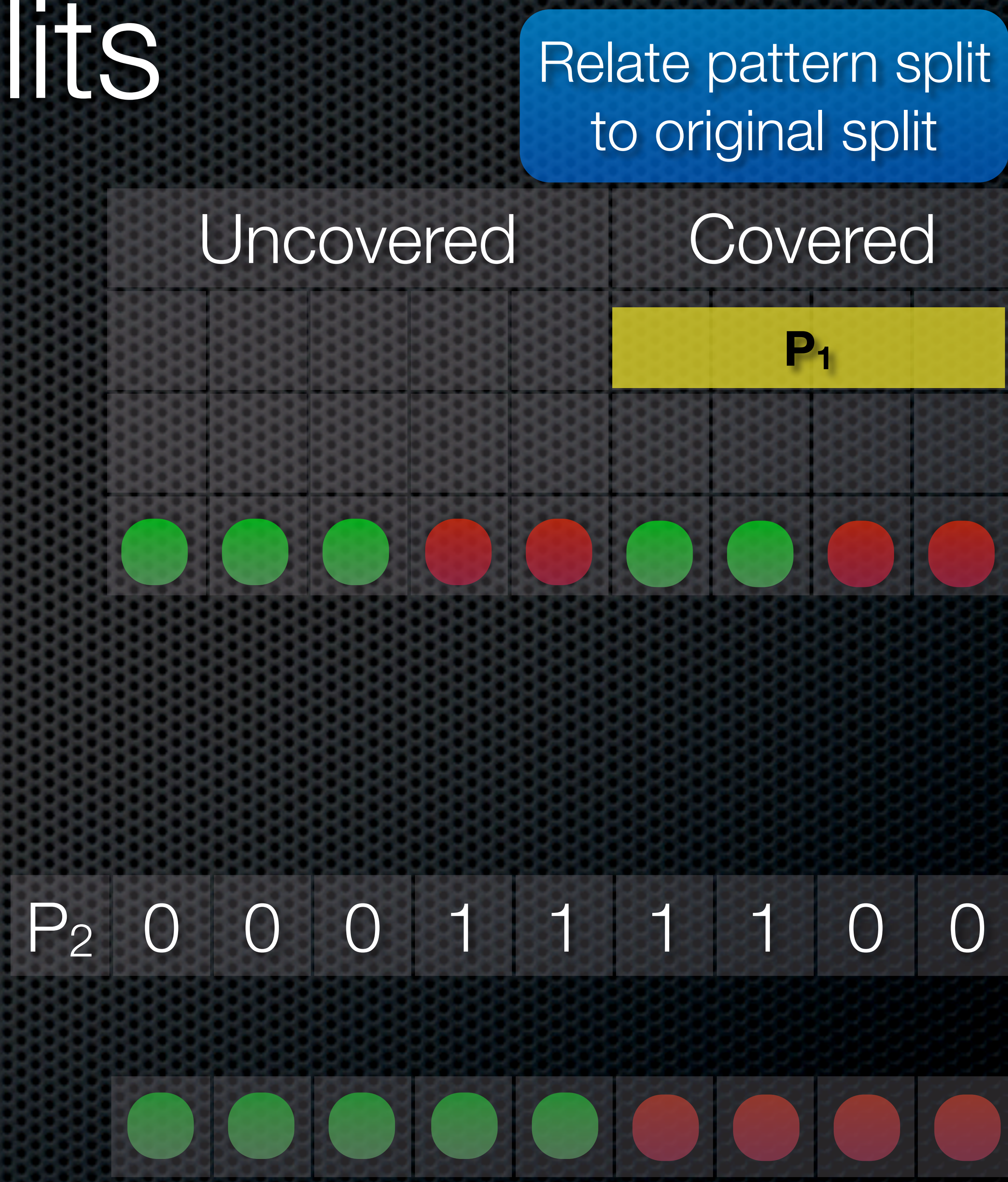
- Pattern absence/ presence becomes new property of data point



Patterns as splits

Imposes new subsets

- Pattern absence/ presence becomes new property of data point

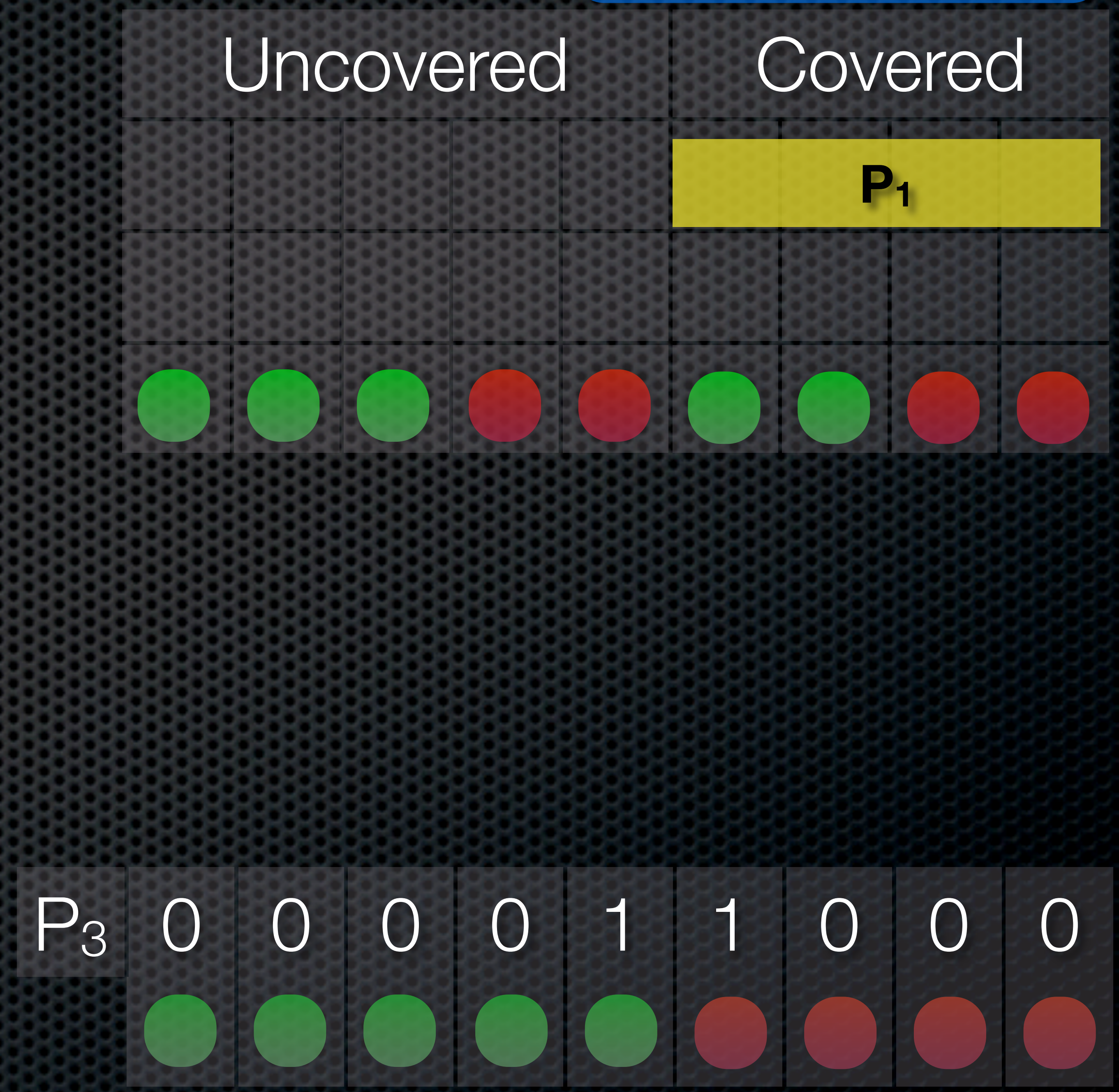


Patterns as splits

Relate pattern split to original split

Imposes new subsets

- Pattern absence/presence becomes new property of data point



Pattern sets as partitions

More than one pattern
⇒ additional presence indicators
⇒ identification of data points with binary vectors
⇒ more numerous splits

P_1	0	0	0	0	1	1	1	1	0
P_2	0	0	0	1	1	1	1	0	0
P_3	0	0	0	0	1	1	0	0	0

Pattern sets as partitions

More than one pattern
⇒ additional presence

indicators
⇒ identification of data

points with binary vectors
⇒ more numerous splits

P_1	0	0	0	0	1	1	1	1	0
P_2	0	0	0	1	1	1	1	0	0
P_3	0	0	0	0	1	1	0	0	0

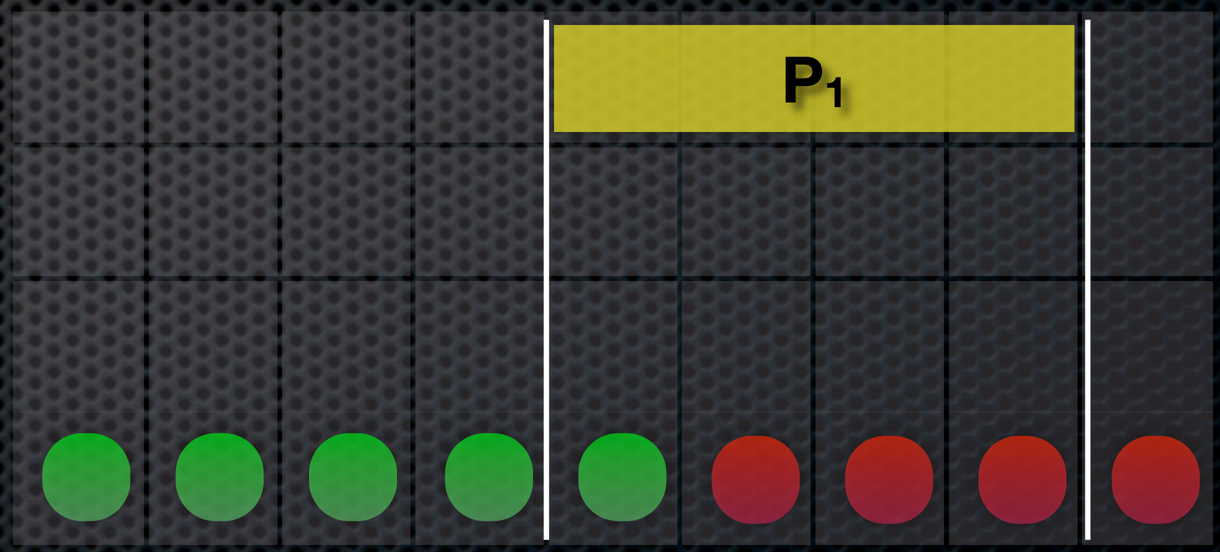
Pattern sets as partitions

More than one pattern
⇒ additional presence

indicators
⇒ identification of data

points with binary vectors
⇒ more numerous splits

P_1	0	0	0	0	1	1	1	1	0
P_2	0	0	0	1	1	1	1	0	0
P_3	0	0	0	0	1	1	0	0	0



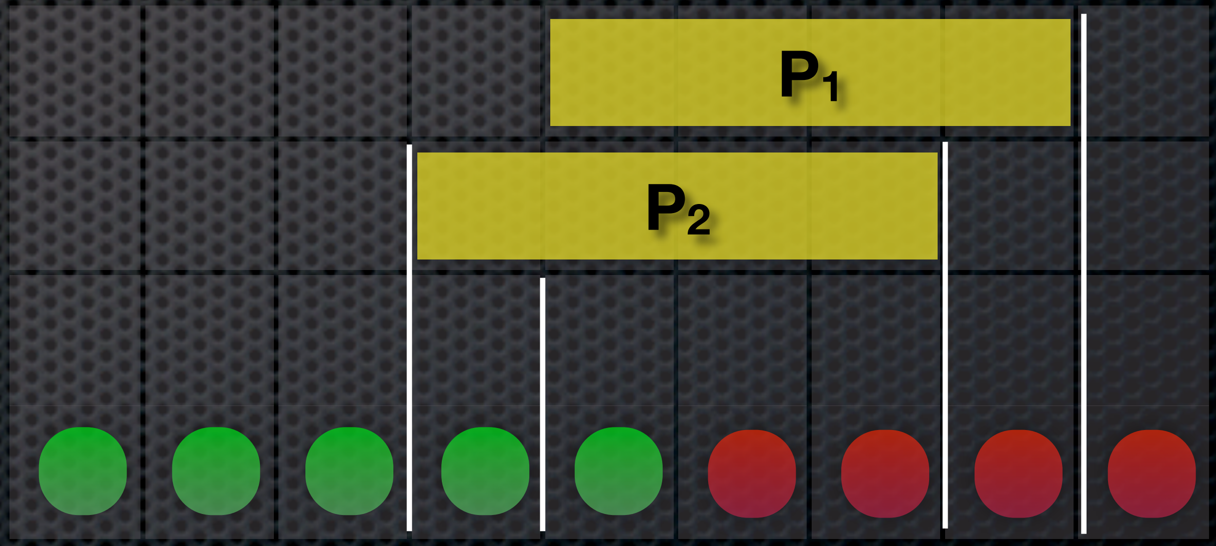
Pattern sets as partitions

More than one pattern
⇒ additional presence

indicators
⇒ identification of data

points with binary vectors
⇒ more numerous splits

P_1	0	0	0	0	1	1	1	1	0
P_2	0	0	0	1	1	1	1	0	0
P_3	0	0	0	0	1	1	0	0	0



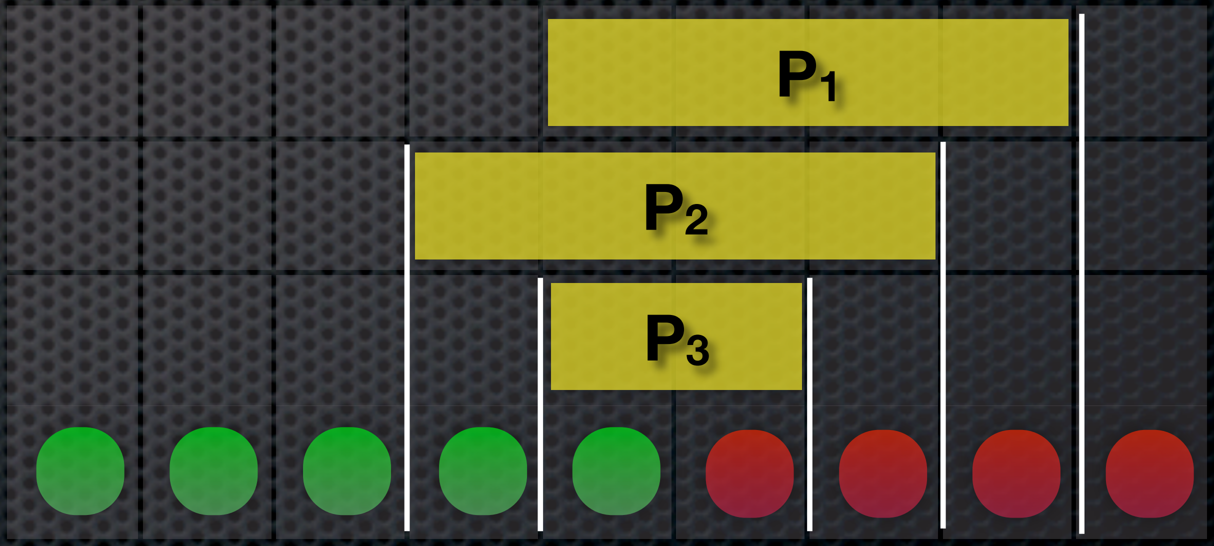
Pattern sets as partitions

More than one pattern
⇒ additional presence

indicators
⇒ identification of data

points with binary vectors
⇒ more numerous splits

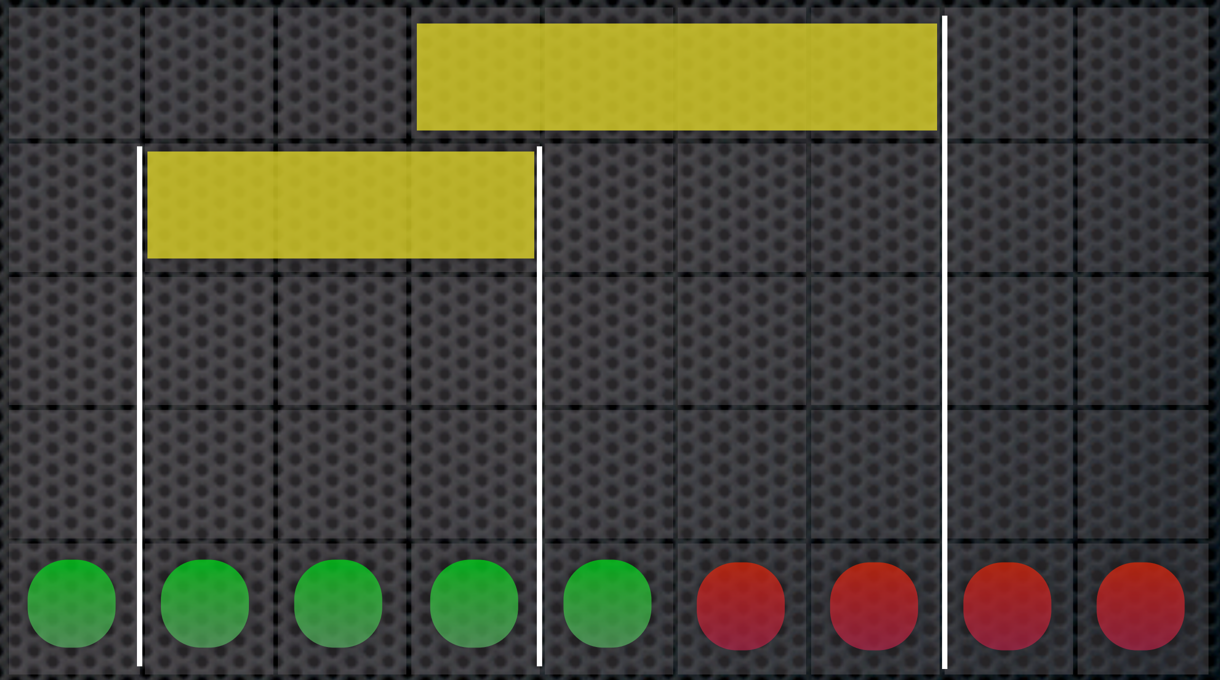
P_1	0	0	0	0	1	1	1	1	0
P_2	0	0	0	1	1	1	1	0	0
P_3	0	0	0	0	1	1	0	0	0



Partitions as Redundancy-Measure

Are new splits created?

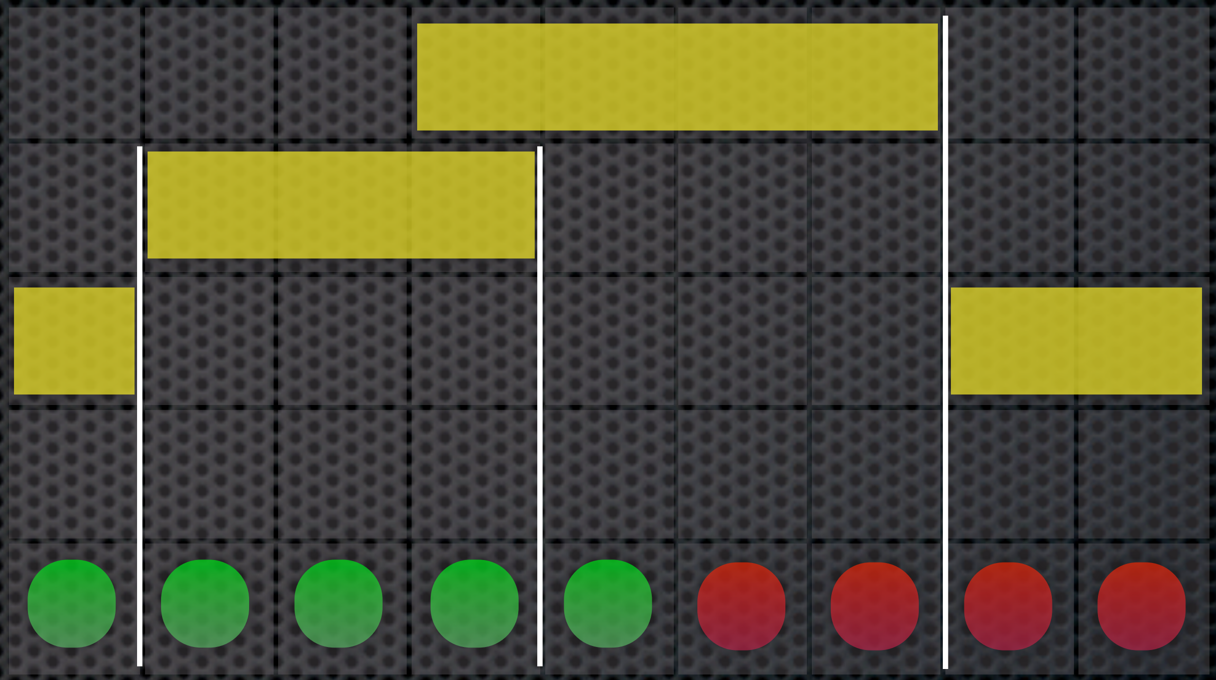
- If not, pattern redundant



Partitions as Redundancy-Measure

Are new splits created?

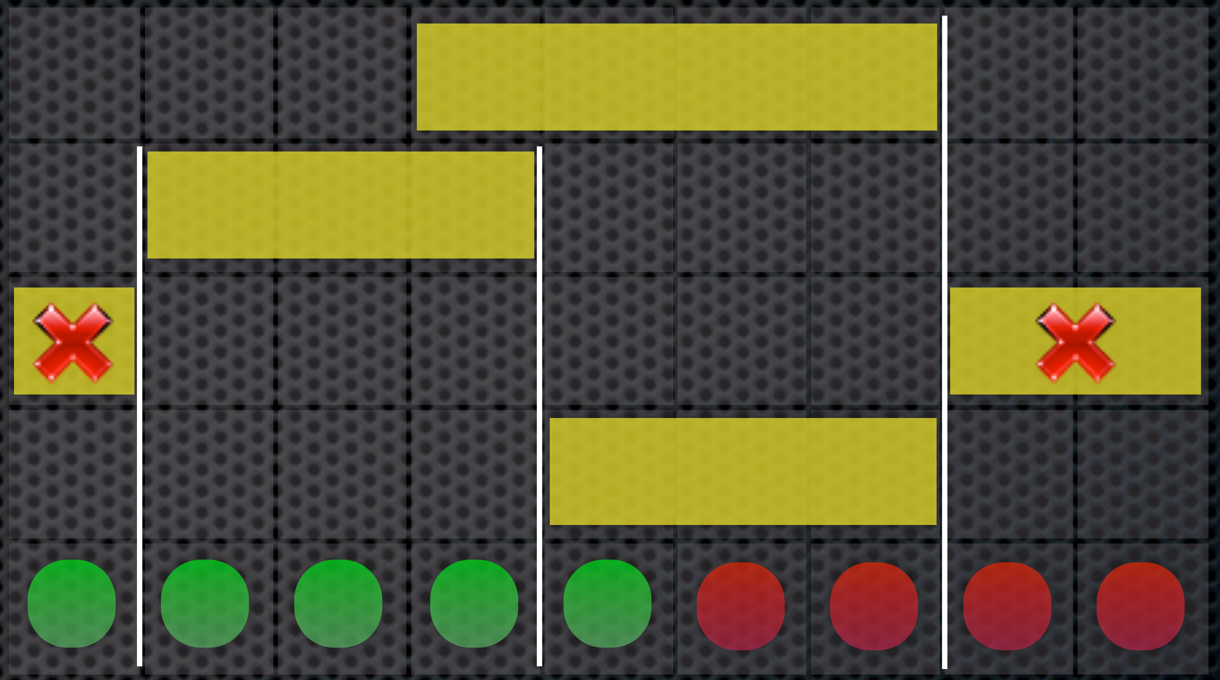
- If not, pattern redundant



Partitions as Redundancy-Measure

Are new splits created?

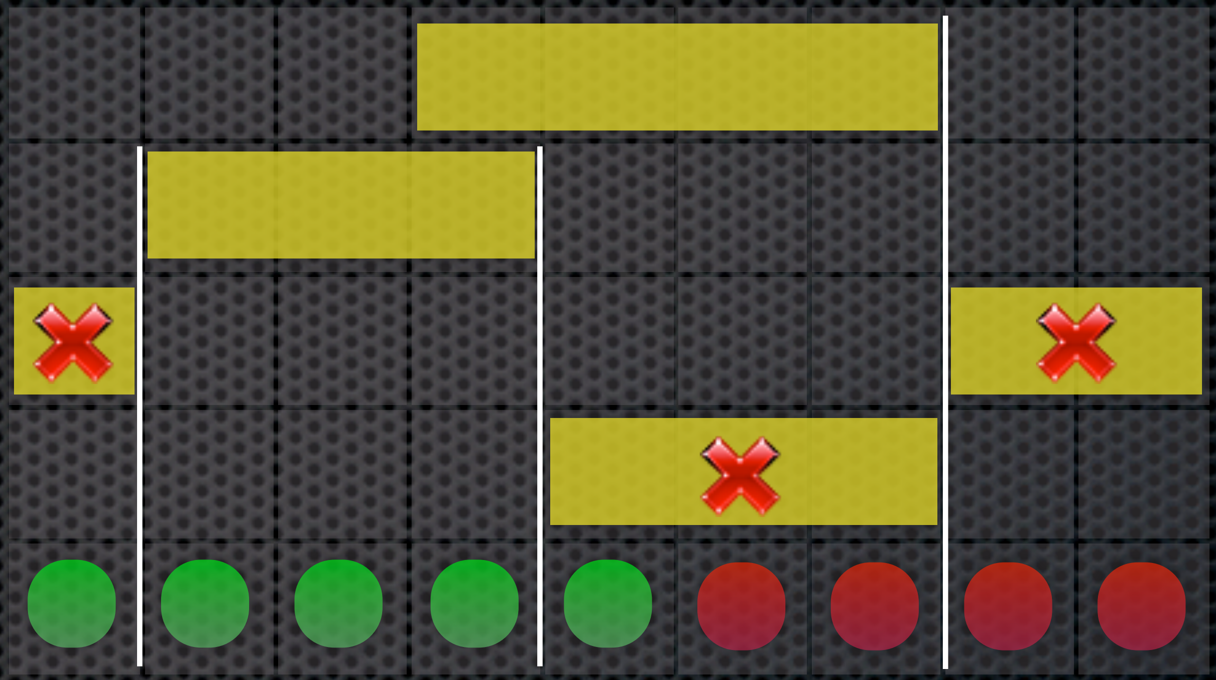
- If not, pattern redundant



Partitions as Redundancy-Measure

Are new splits created?

- If not, pattern redundant



Putting it together

Goal: “optimal” pattern set for given task

- Globally optimal mostly impossible
- Also, there’s over-fitting

Locally optimize supervised measure

- For individual pattern

Accuracy
Correlation
Correspondences

Refine partition

- To avoid redundancy, encourage diversity

Choose next pattern

Post-Processing (CBA)

Measure: Confidence

Optimization: Locally

Partition refinement: Globally, implicit

Fixed order on patterns

Sequentially processed

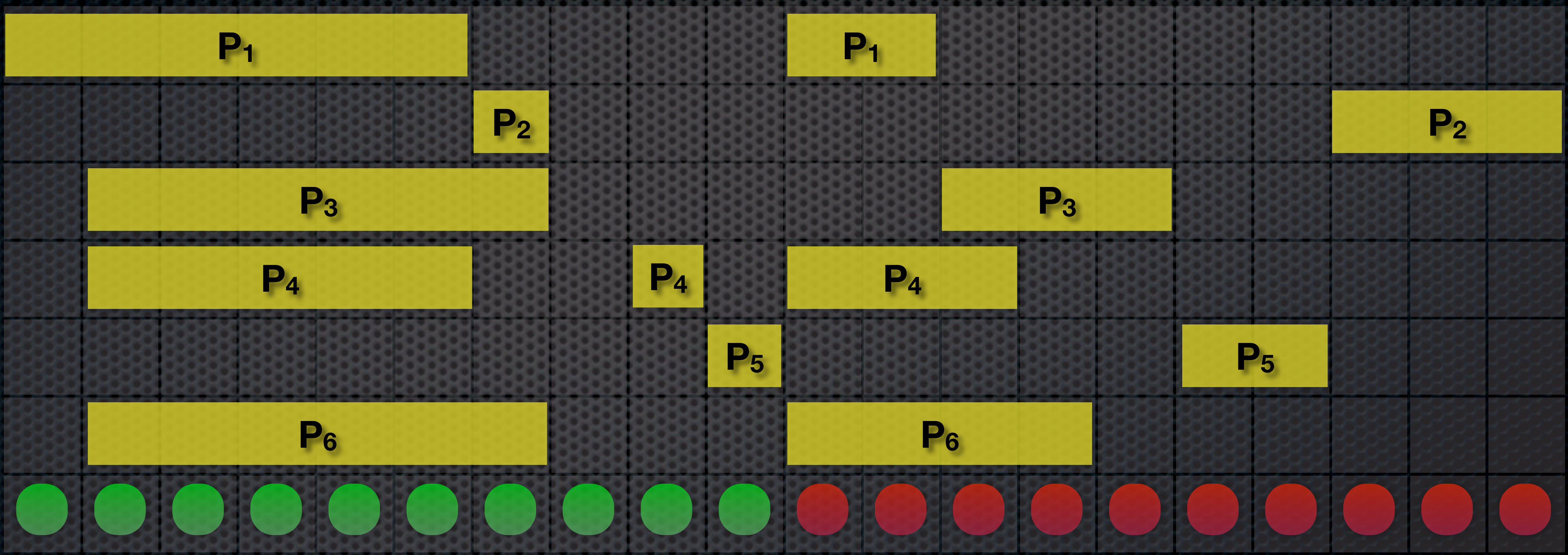
Only consider **uncovered** data points

Patterns **have to** classify correctly

How does it look?

Ordered by descending confidence

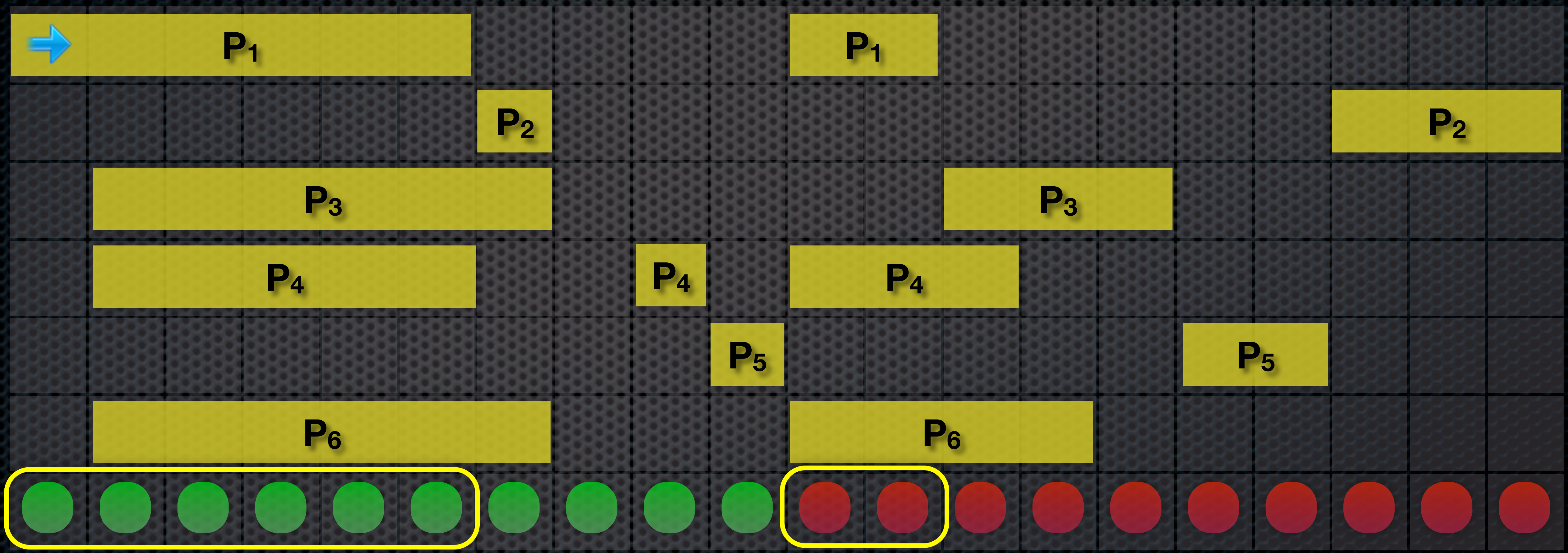
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

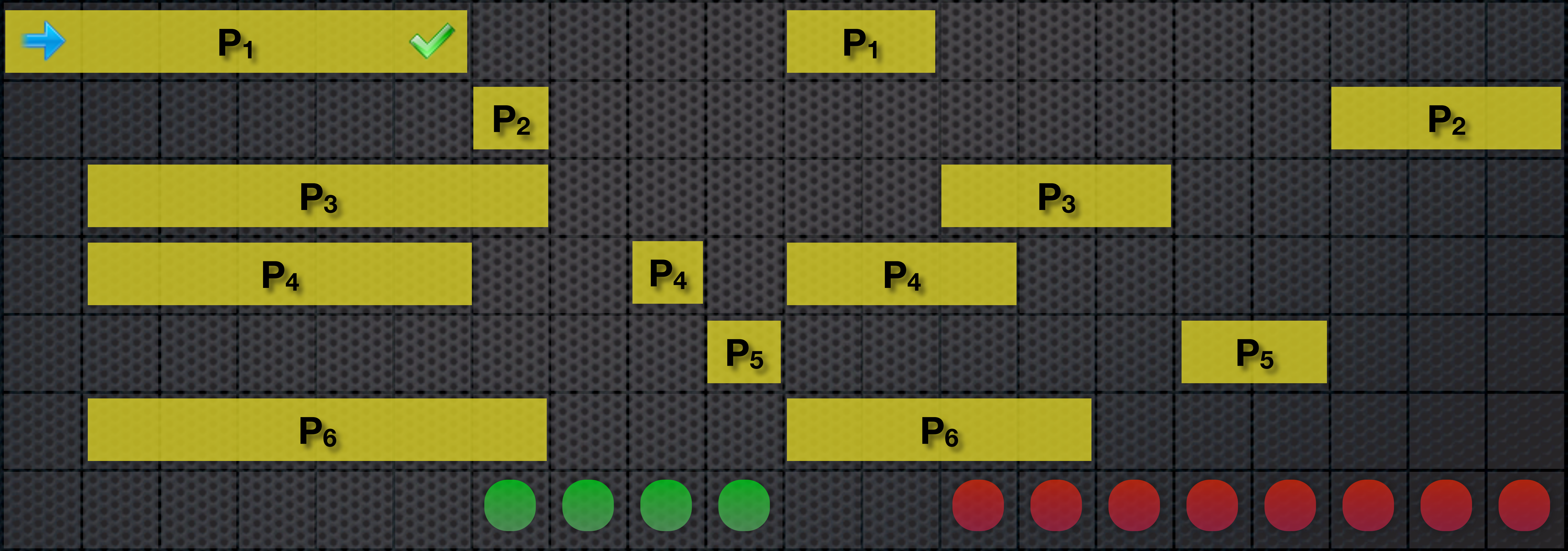
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

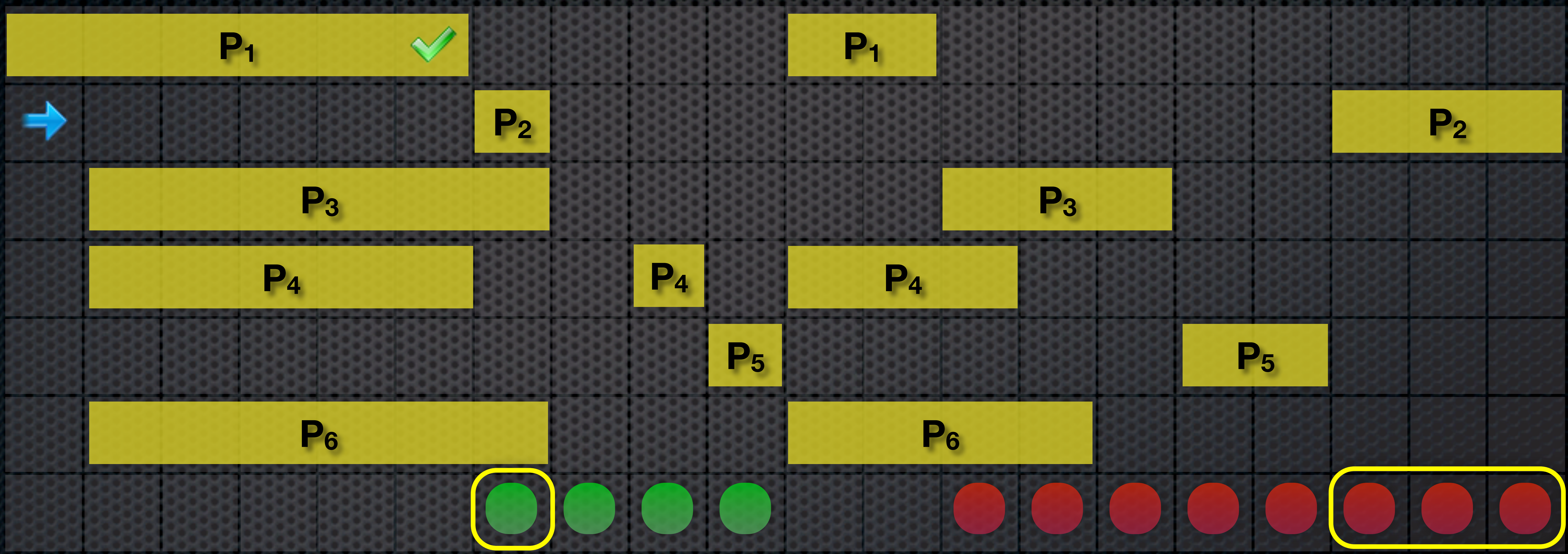
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

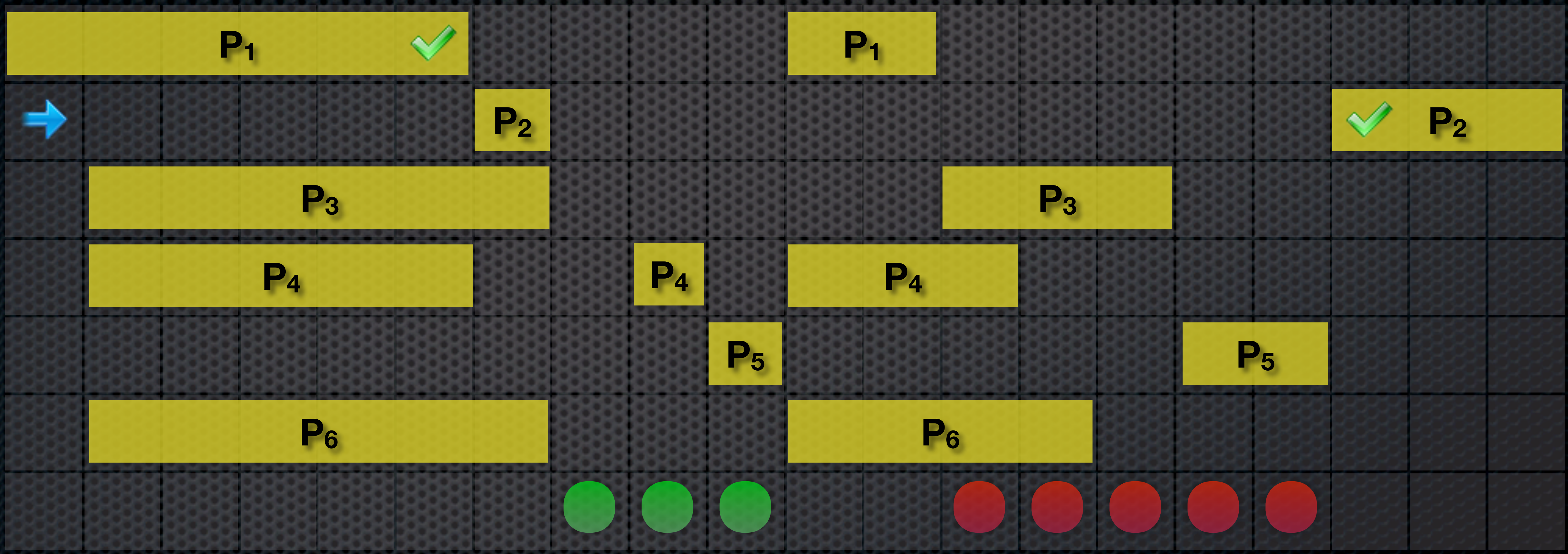
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

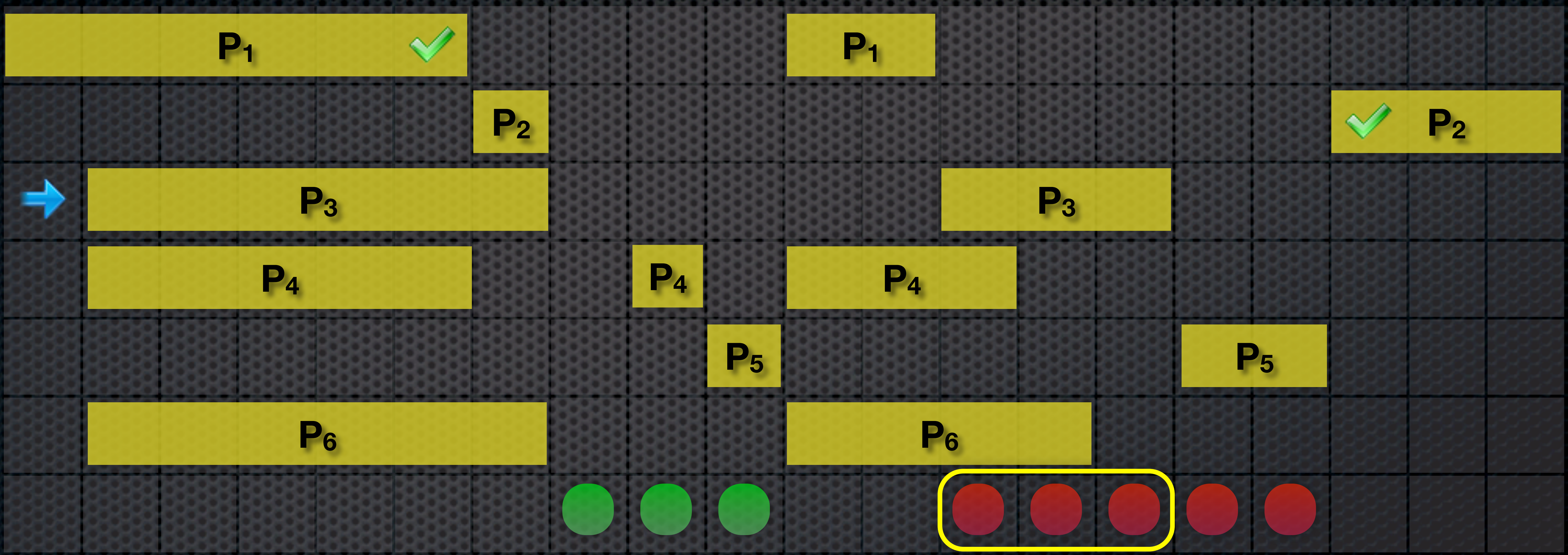
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

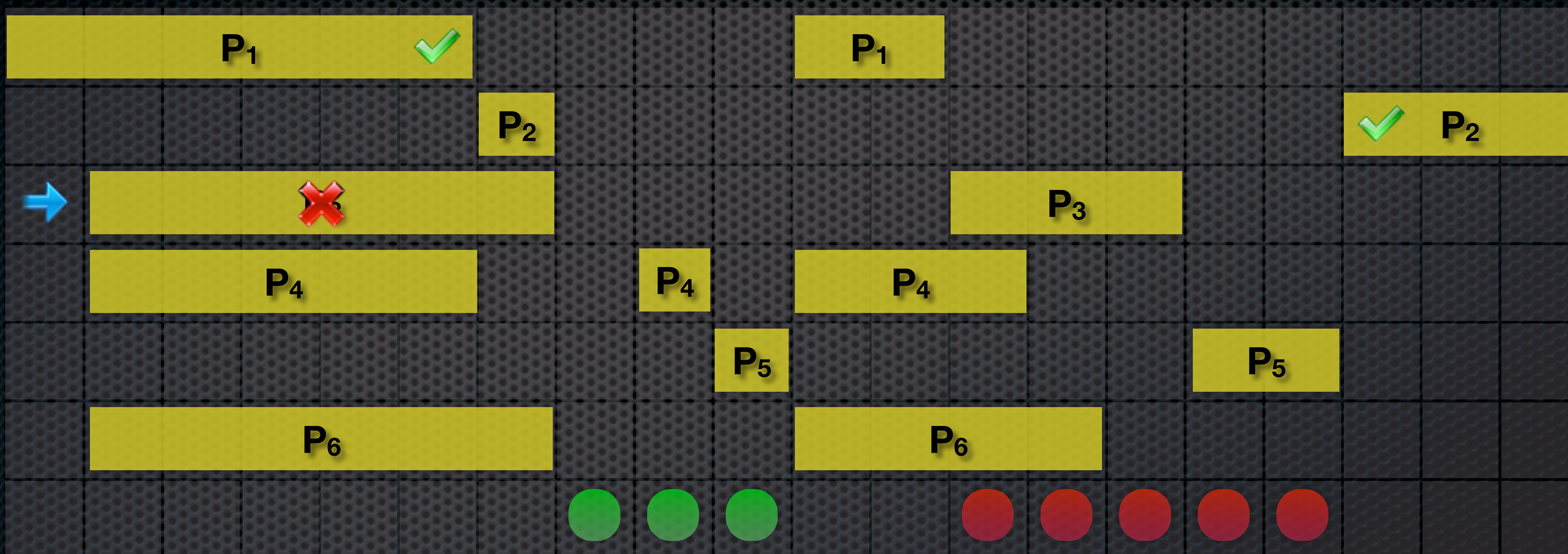
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

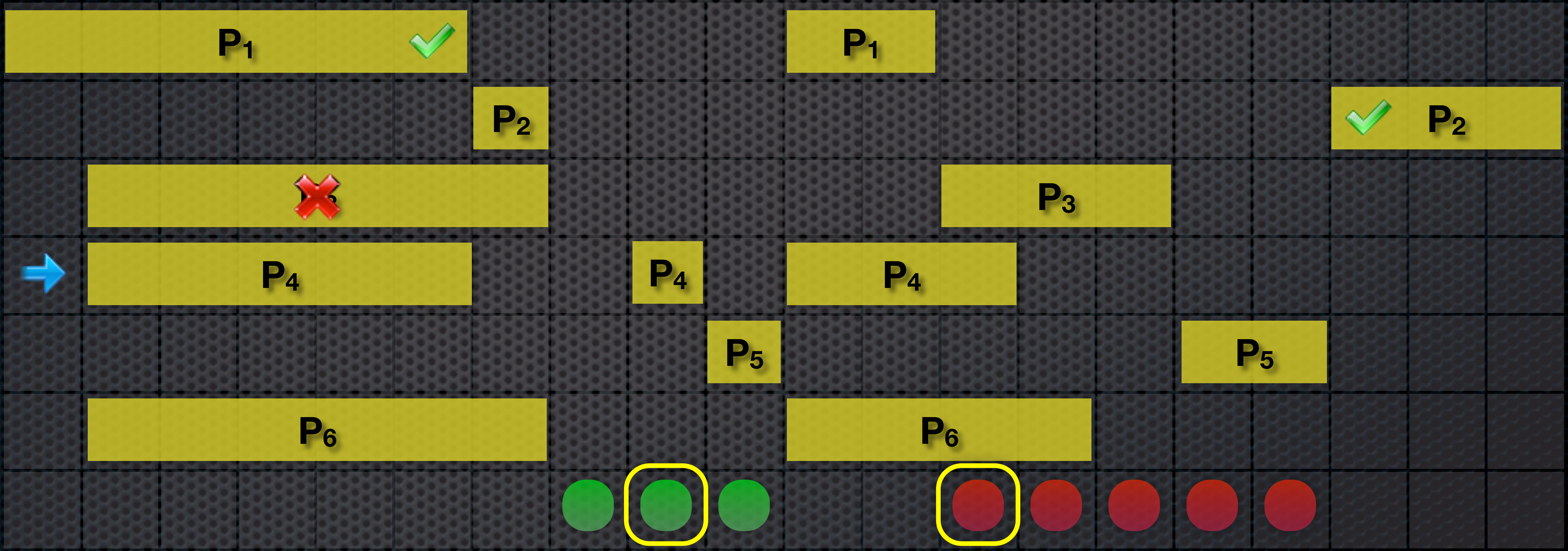
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

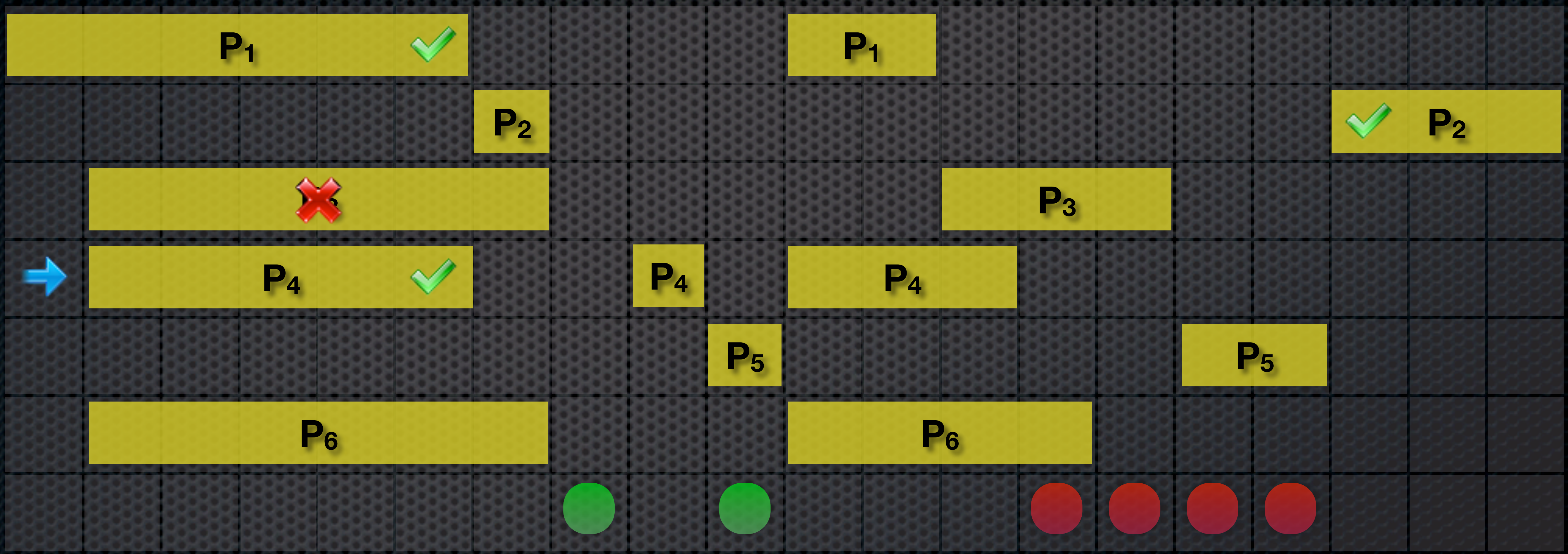
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

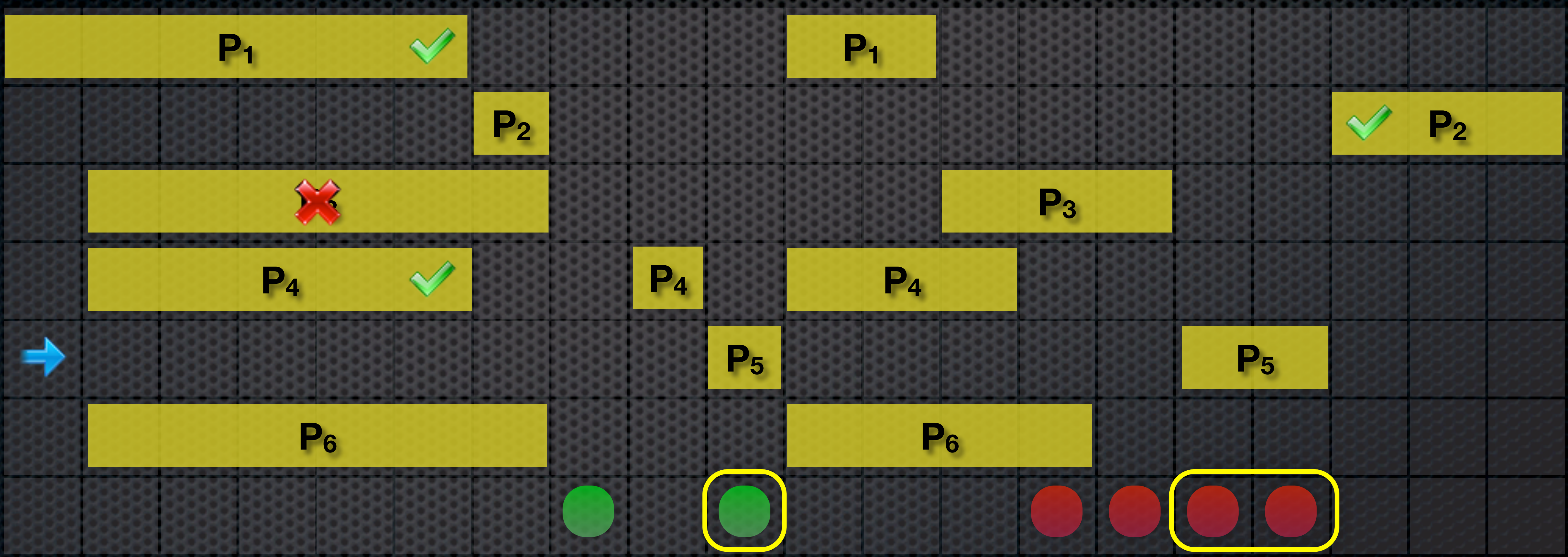
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

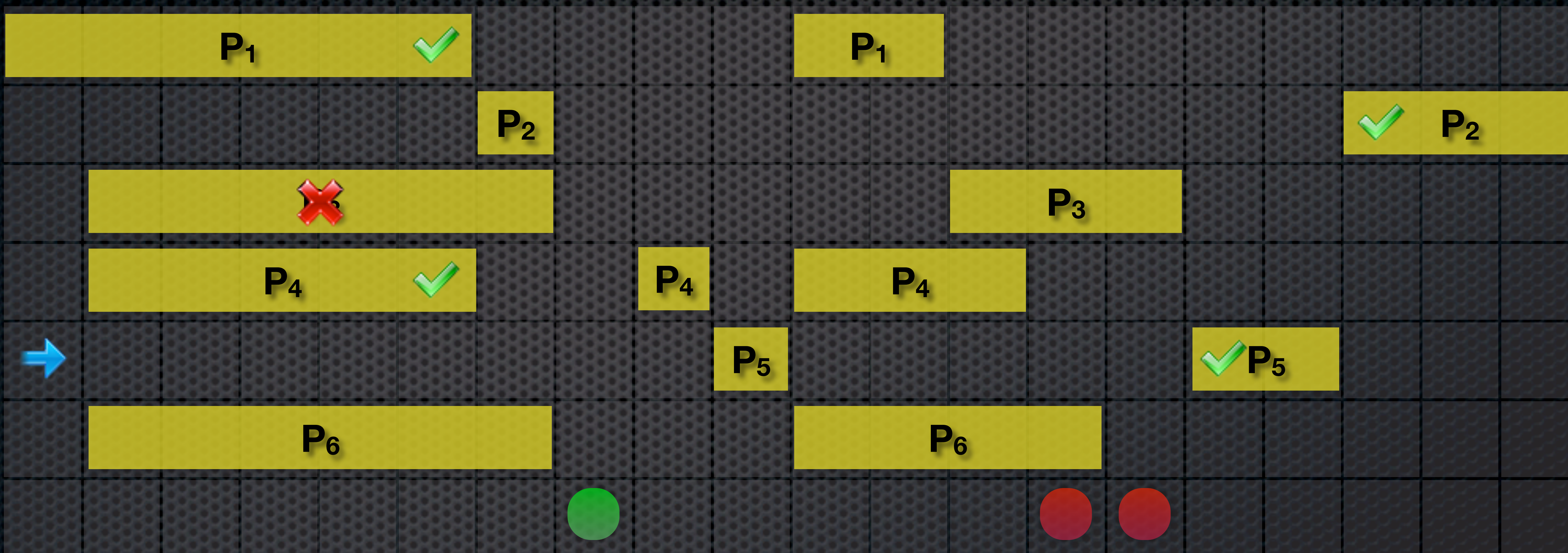
- Same confidence → by descending support



How does it look?

Ordered by descending confidence

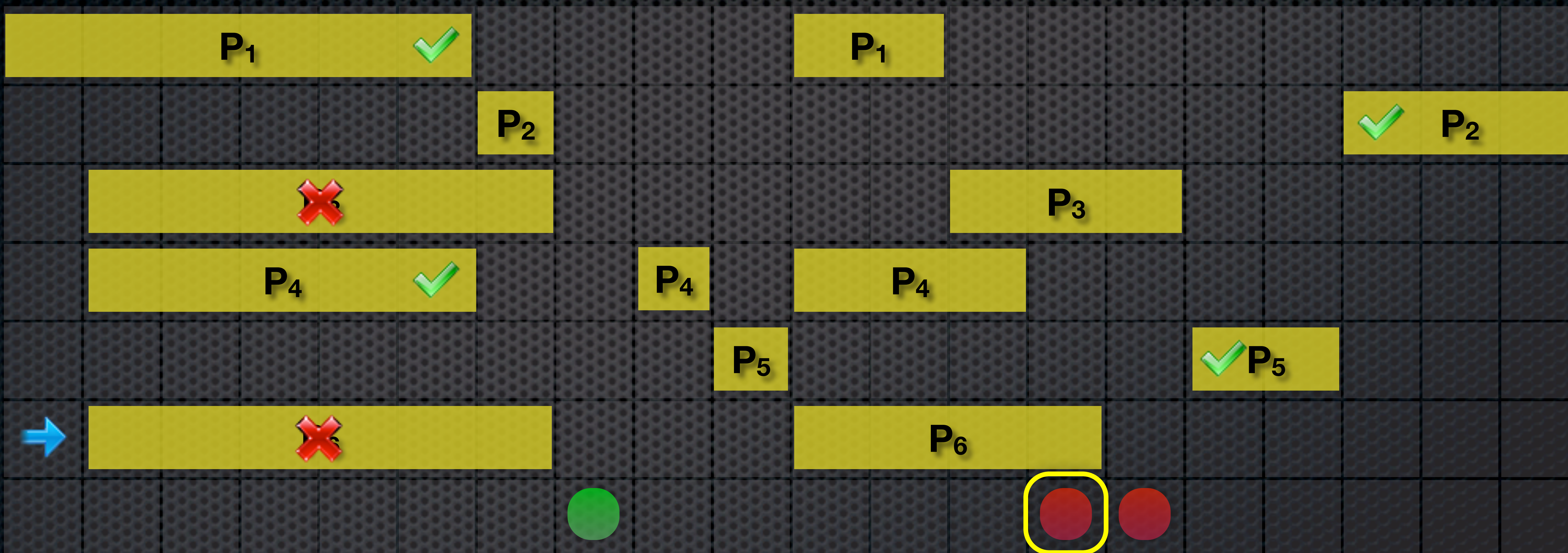
- Same confidence → by descending support



How does it look?

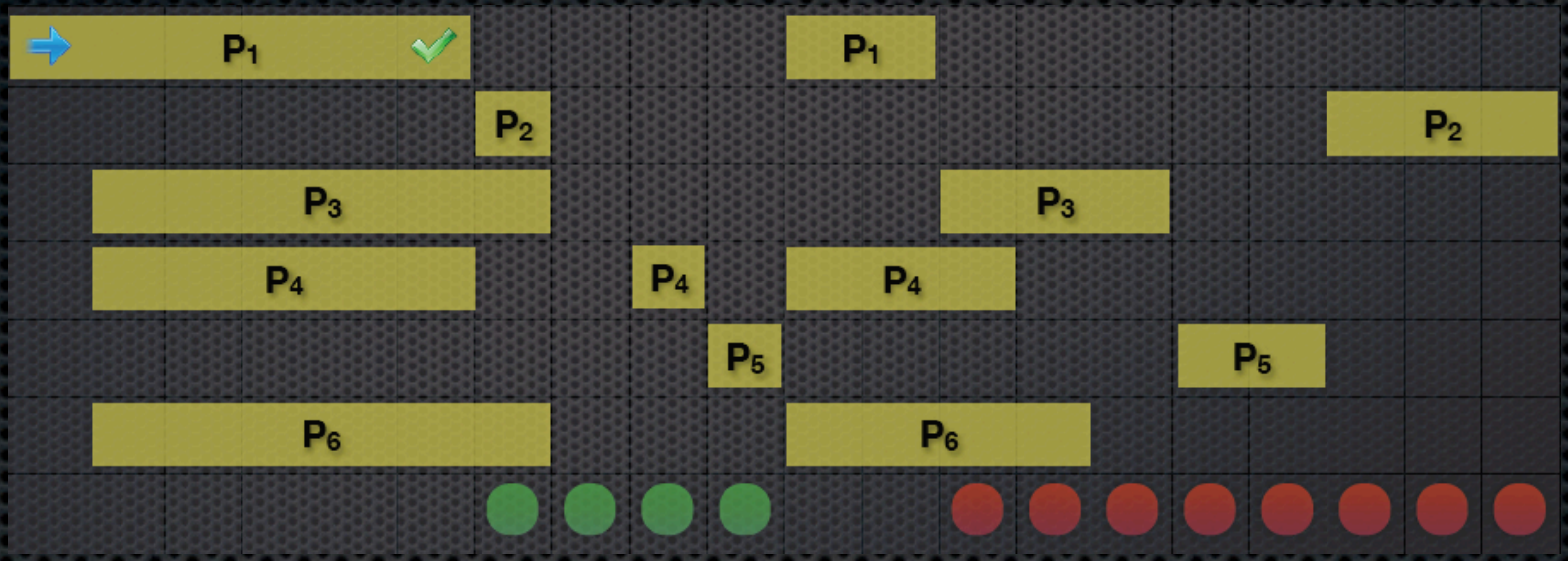
Ordered by descending confidence

- Same confidence → by descending support

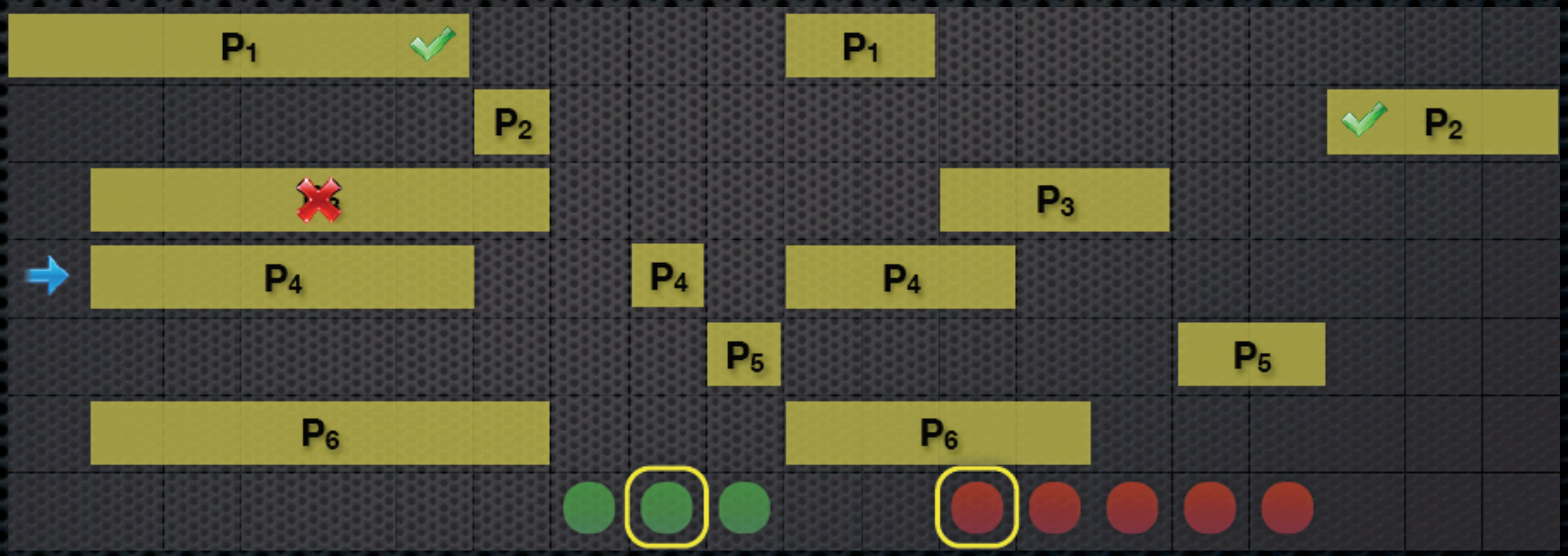


(Potential) Problems

Post-processing can miss out on interesting patterns



Fixed order doesn't take changes in partition into account



Remind you of something?

Well-known Machine Learning technique: Sequential covering

- Used to learn classification rules
- Find very accurate rule
- Remove covered examples
- Learn on the remains

Iterative Mining!

1. Mine “optimal” pattern
2. Refine partition
3. Re-iterate

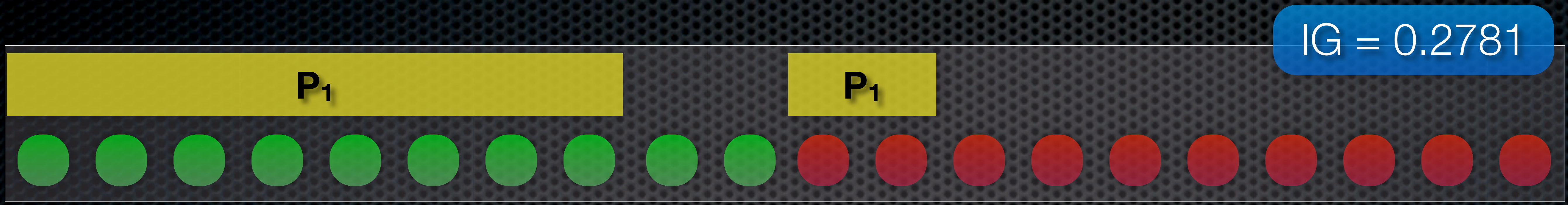
DTM - Decision Tree Mining

Measure: Information Gain (as in decision trees)
Optimization: Locally
Split: Locally (as in decision trees), explicit

Mine pattern maximizing InfoGain
Use pattern to split data on which it was mined in 2 subsets
Reiterate on subsets

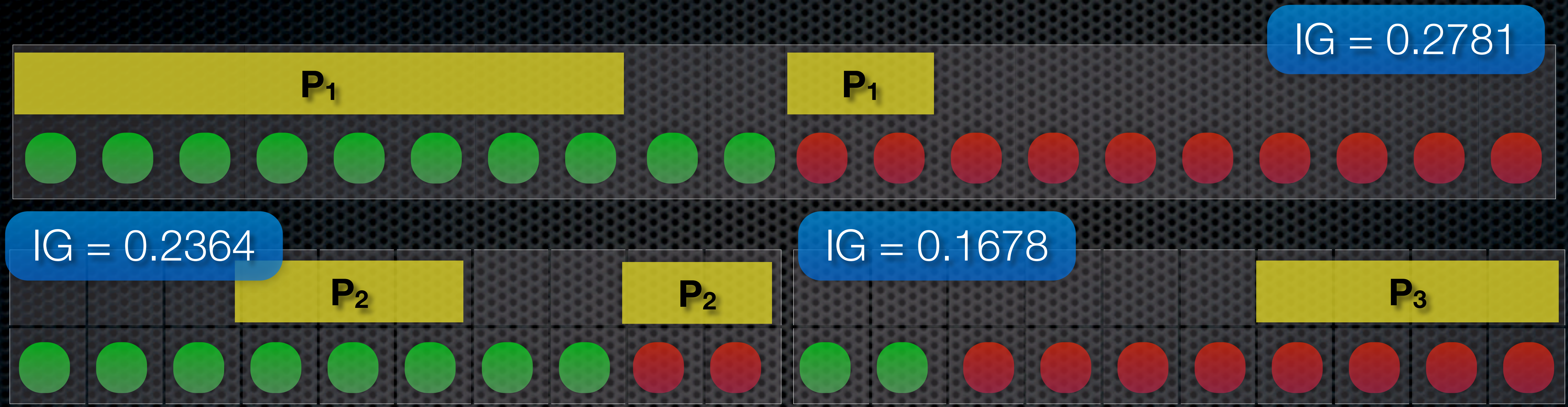
How does it look?

How does it look?

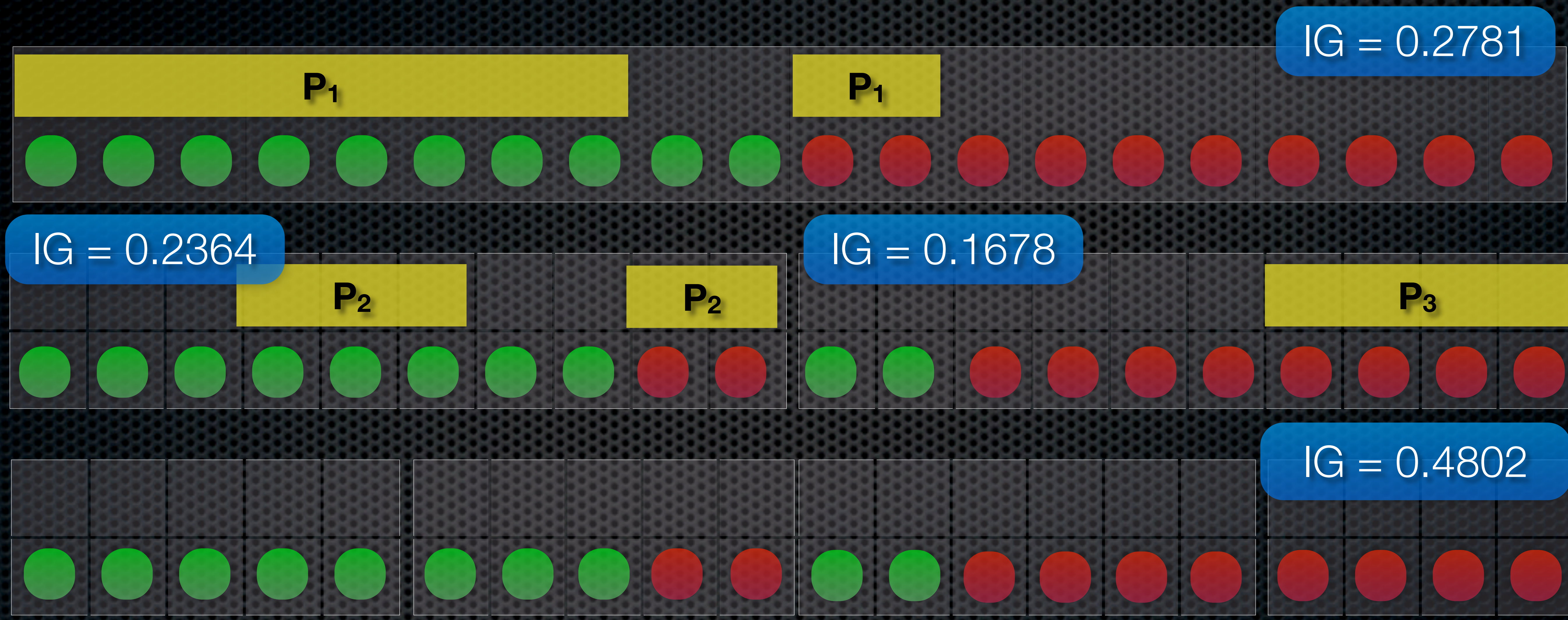


How does it look?

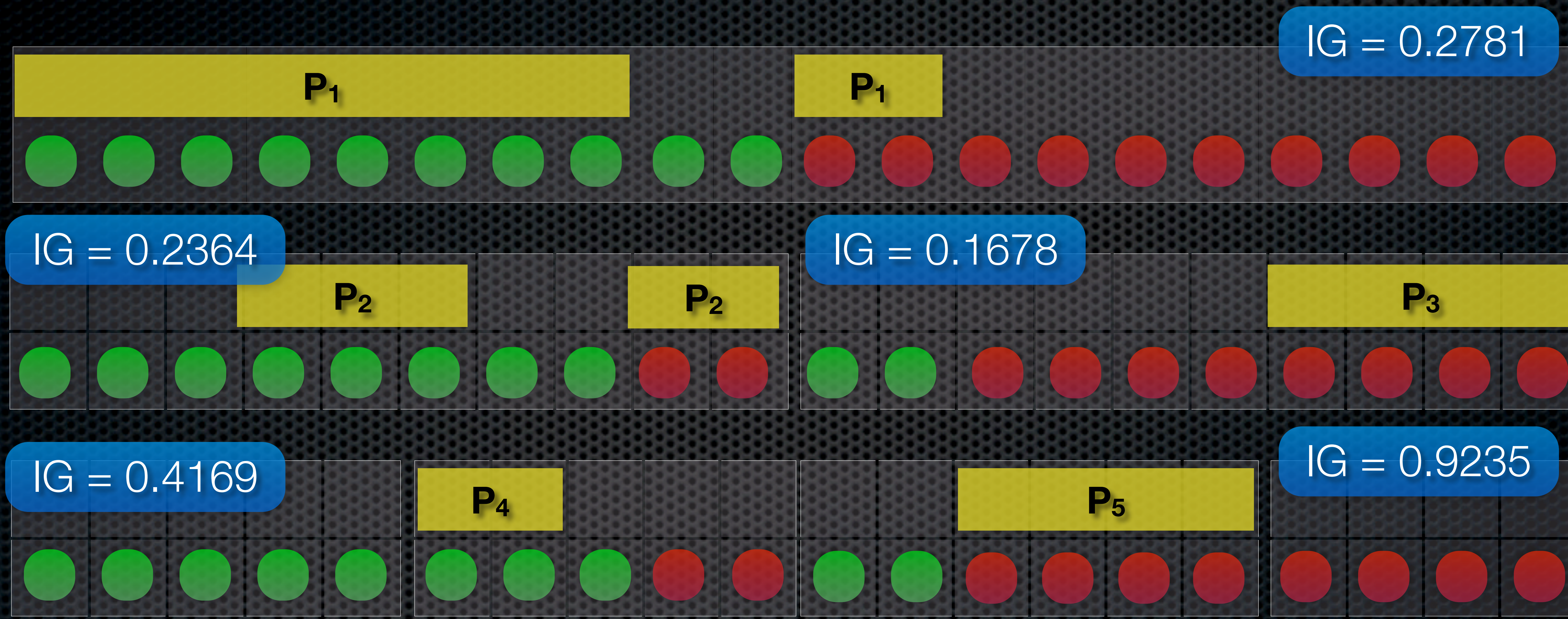
How does it look?



How does it look?

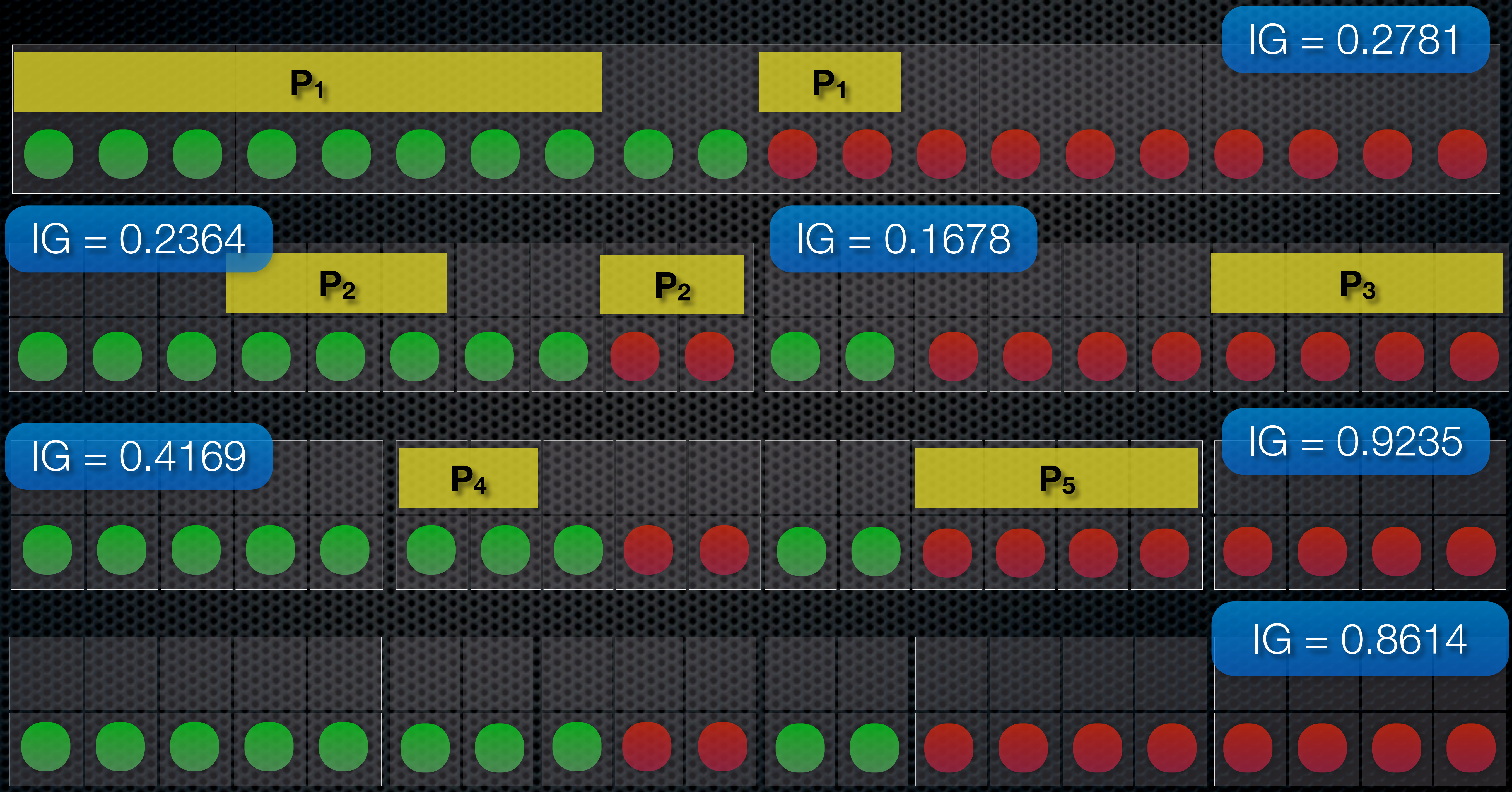


How does it look?



How does it look?

How does it look?



Pros and Cons

Single patterns do not have to be overly accurate

Reuses data unless purified

- Gradual refinement of description possible
- Over-fitting possible

Gradually smaller subsets

- Allows parallelization
- Harder cases, fewer candidates

Local measure optimization

- Less reliable evaluation individual patterns
- Many patterns, can be (partially) redundant

Partition refinement only locally

- Discards information about pattern effects
- Increases uncertainty about contribution single pattern

fCork

Measure: Correspondences

Optimization: Globally

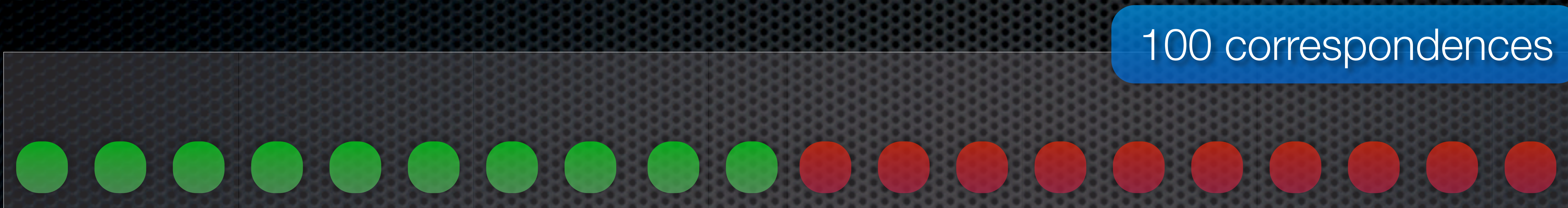
Partition refinement: Globally, implicit

Mine pattern reducing correspondences best

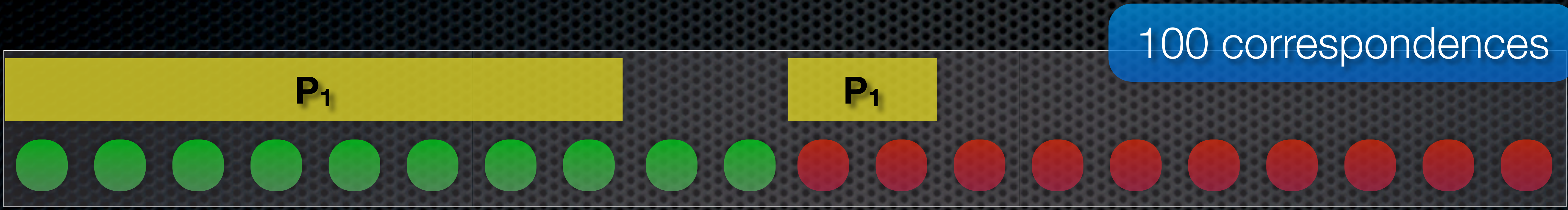
Remove “pure” data points

Re-iterate

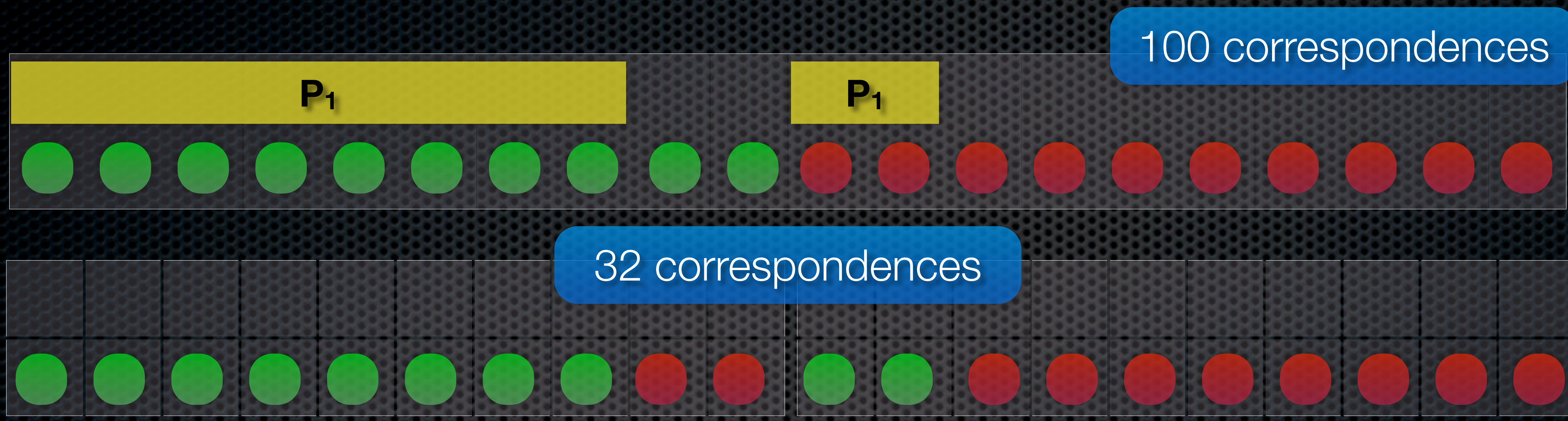
How does it look?



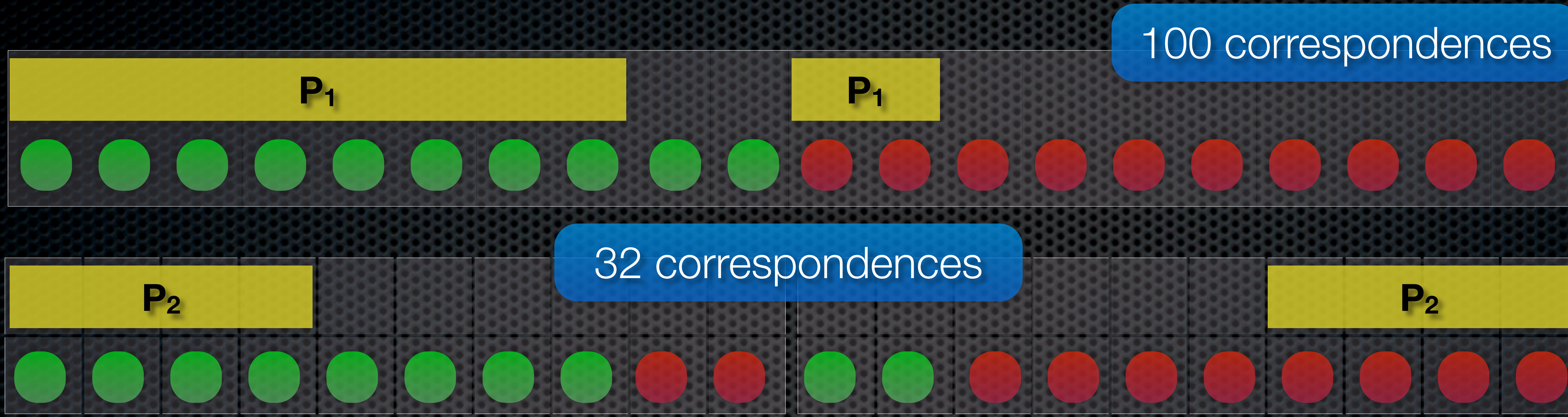
How does it look?



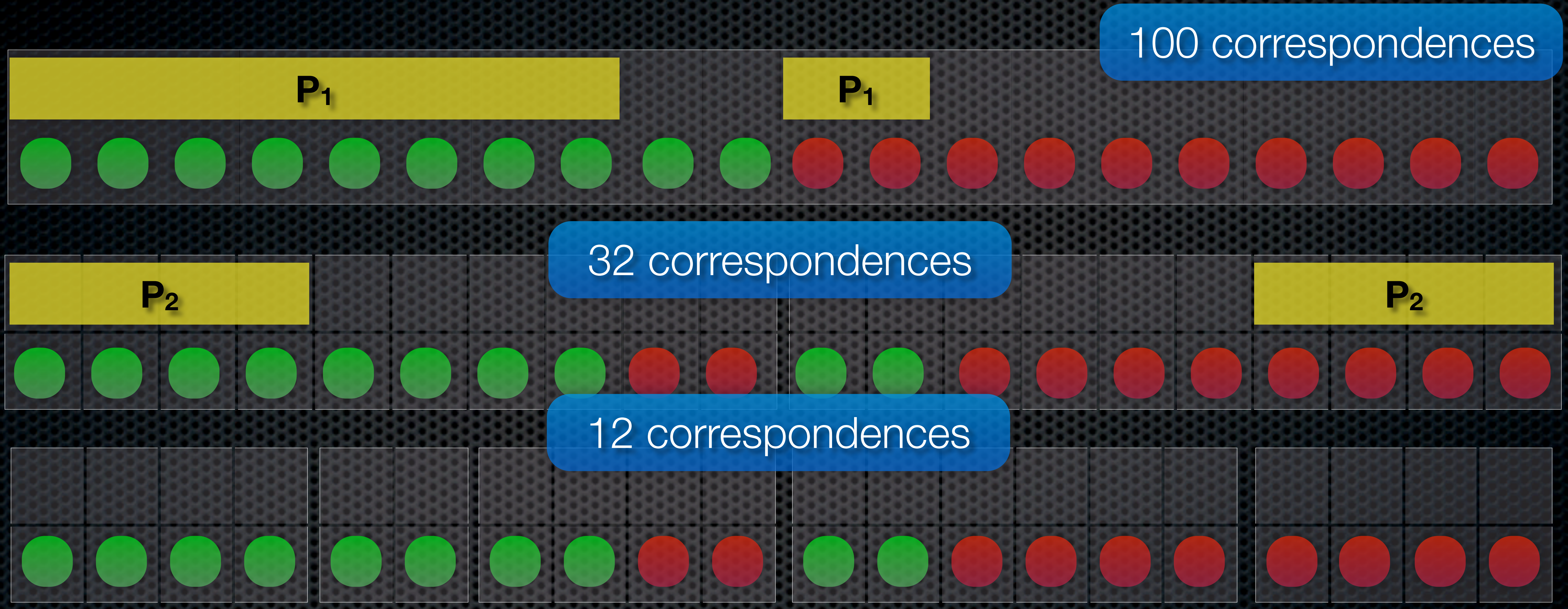
How does it look?



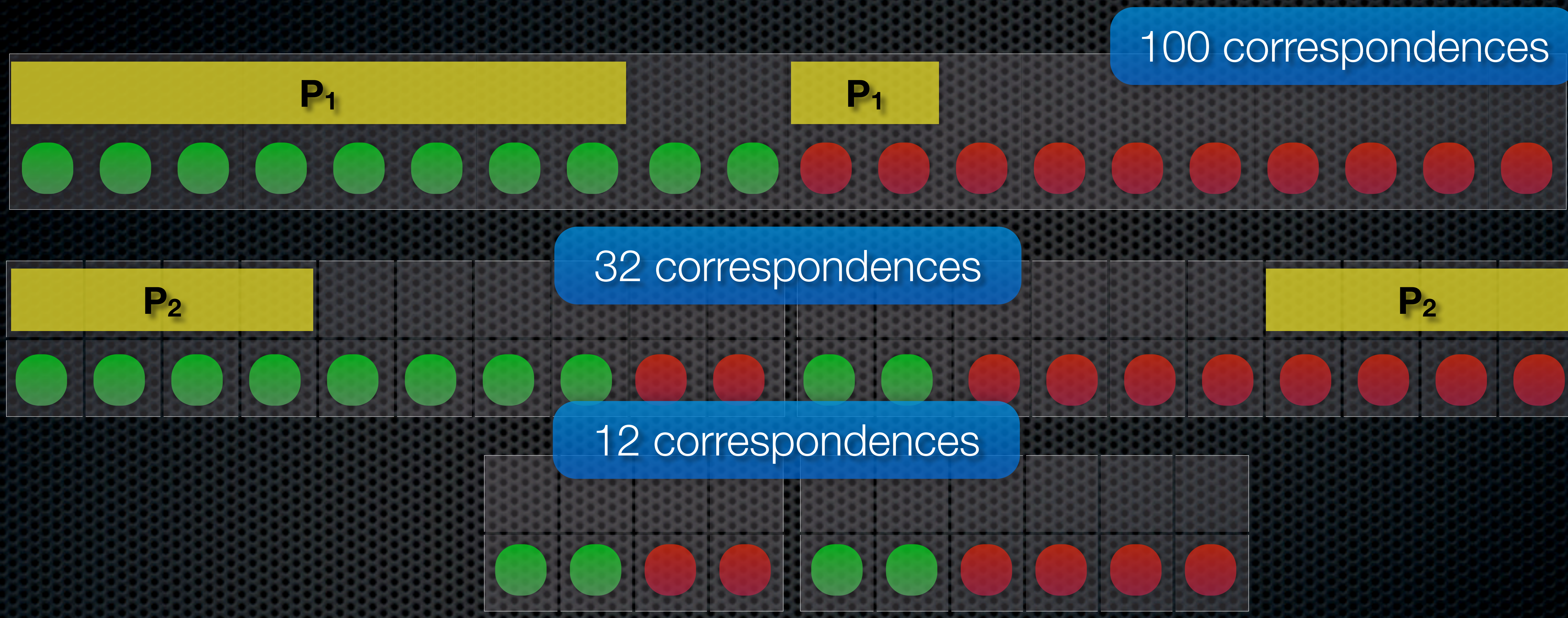
How does it look?



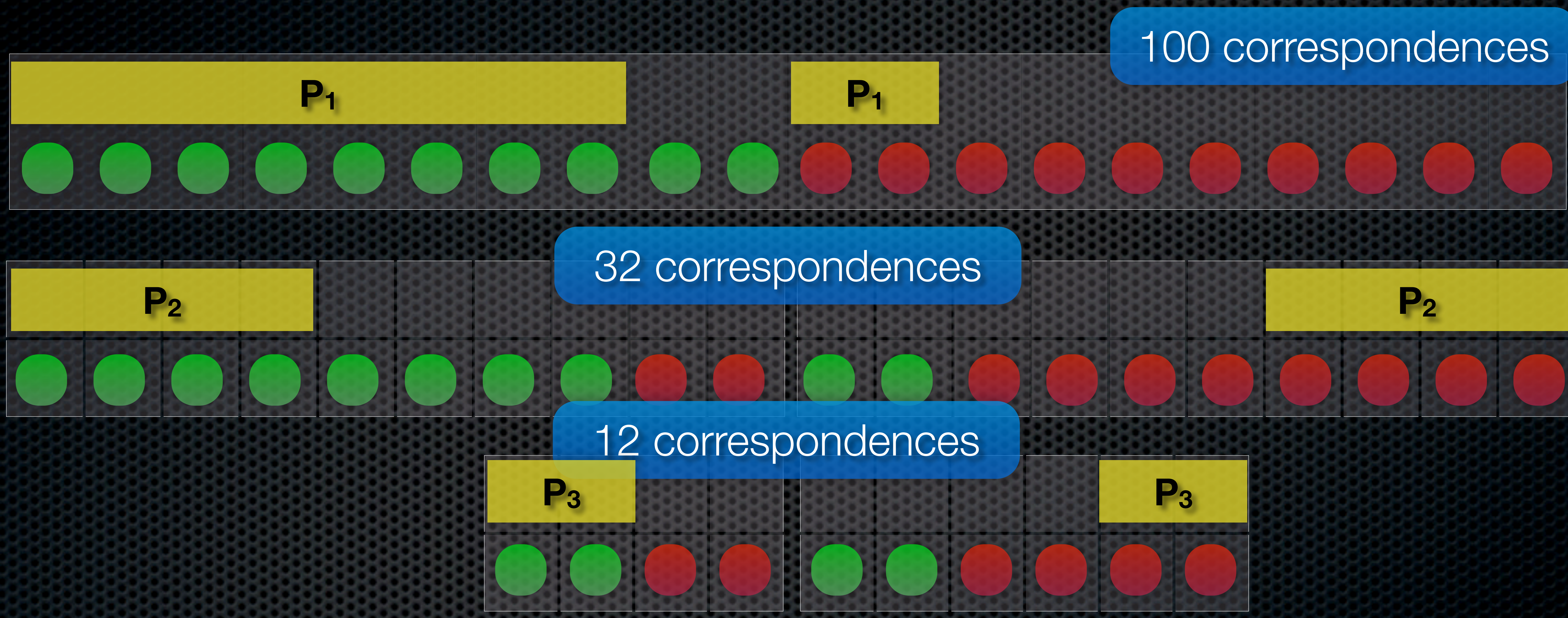
How does it look?



How does it look?

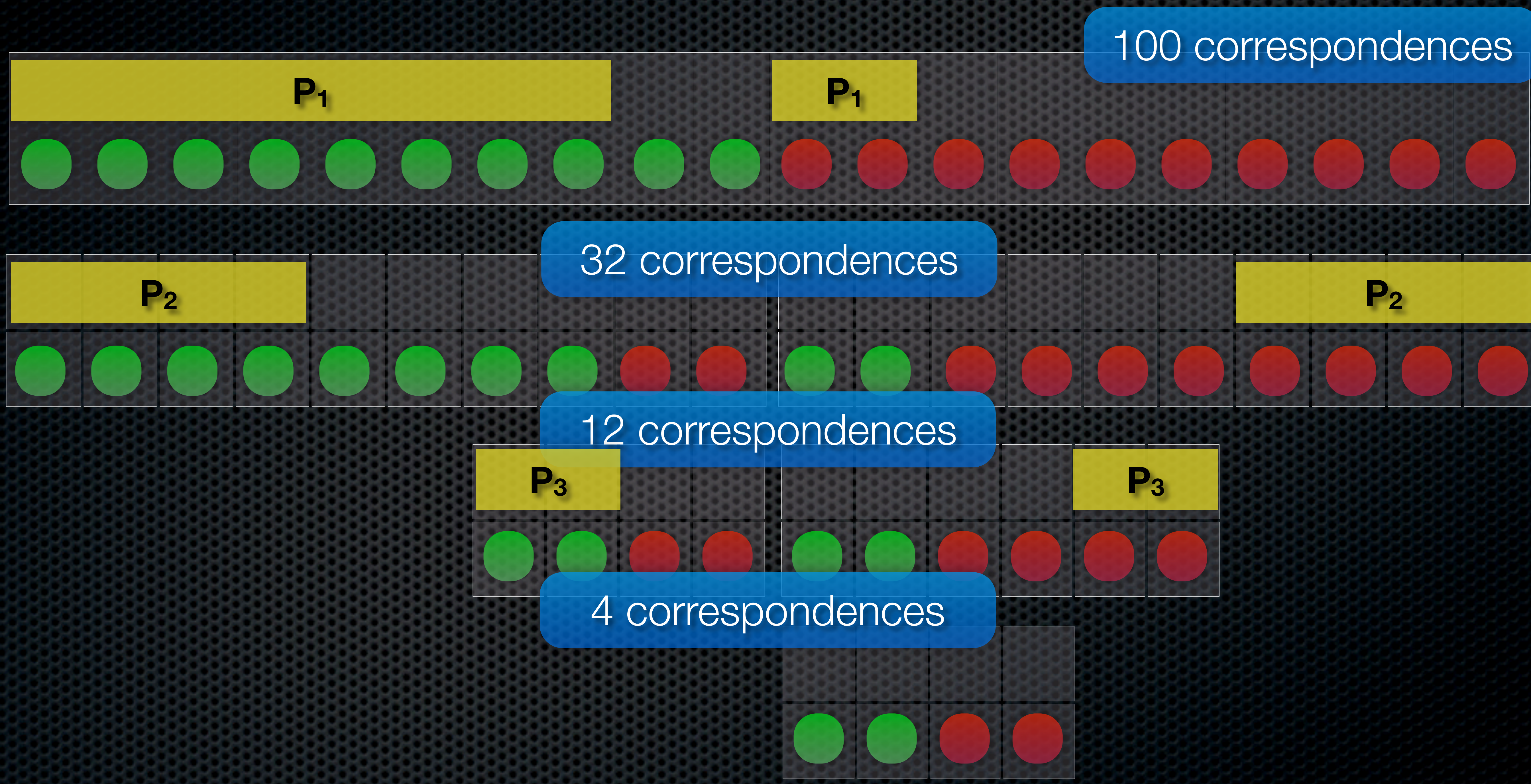


How does it look?

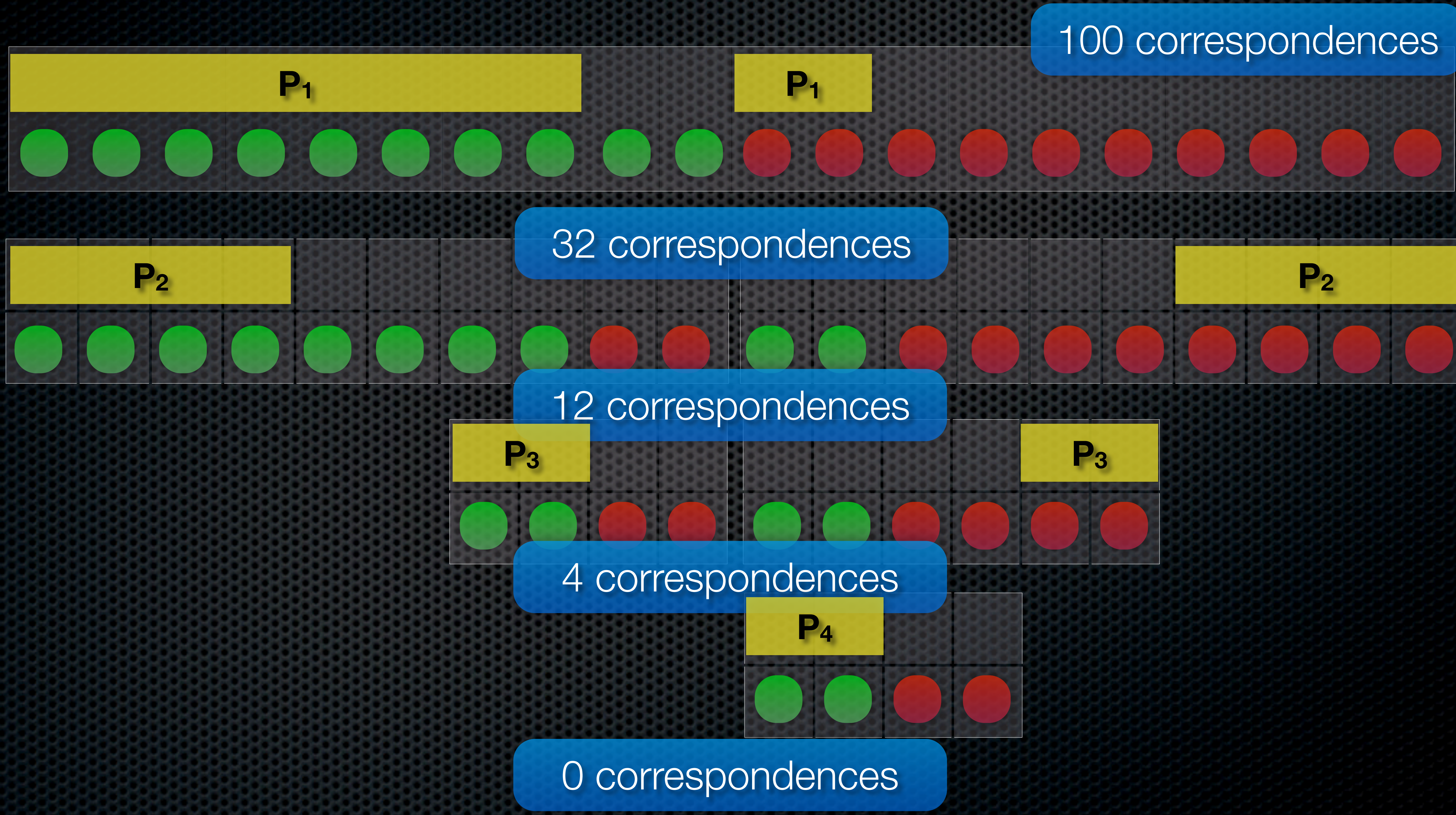


How does it look?

How does it look?



How does it look?



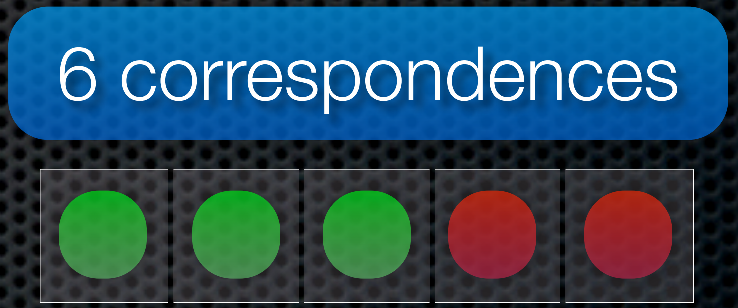
Pros and Cons

WYSIWYG

- Global optimization allows concrete evaluation of pattern contribution
- Due to submodularity

Fewer data in later runs

- Harder cases, less candidates



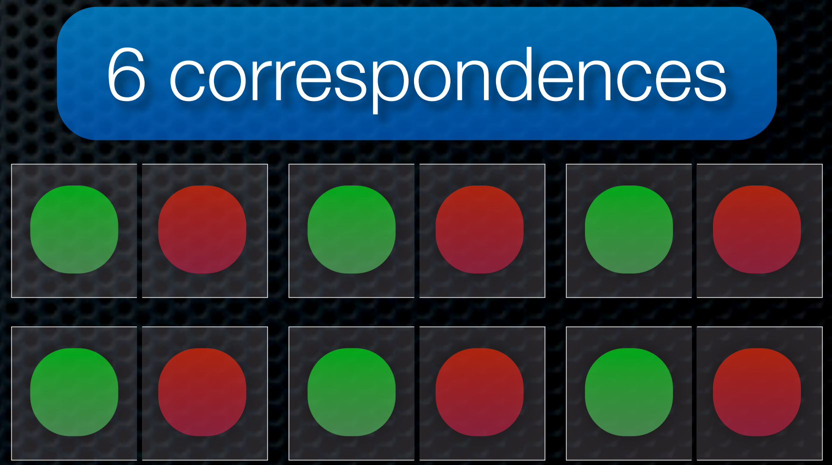
Global partition refinement

- Fewer patterns

Slower for individual patterns

- Due to global evaluation

Correspondences ≠ correspondences



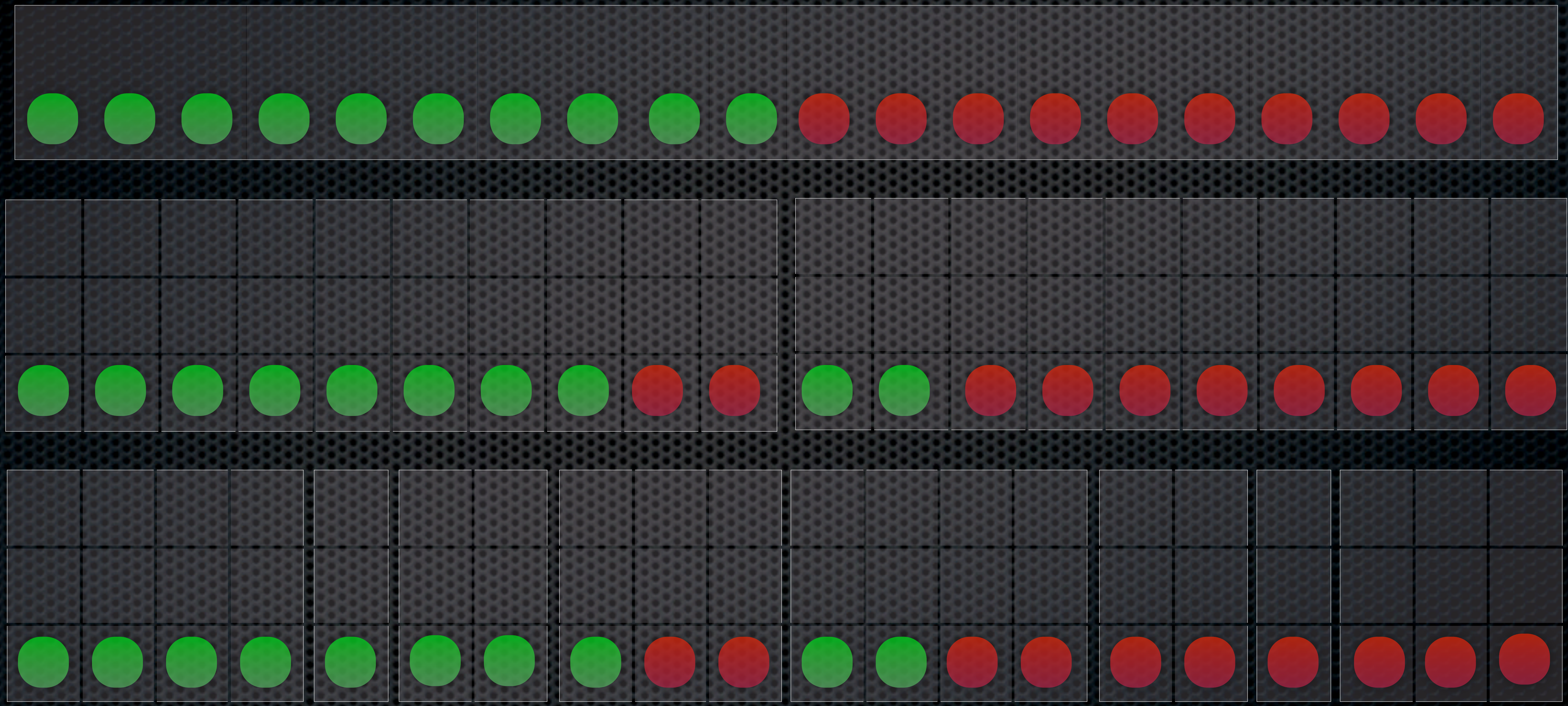
ReMine

Let's build a hybrid!
(recent work)

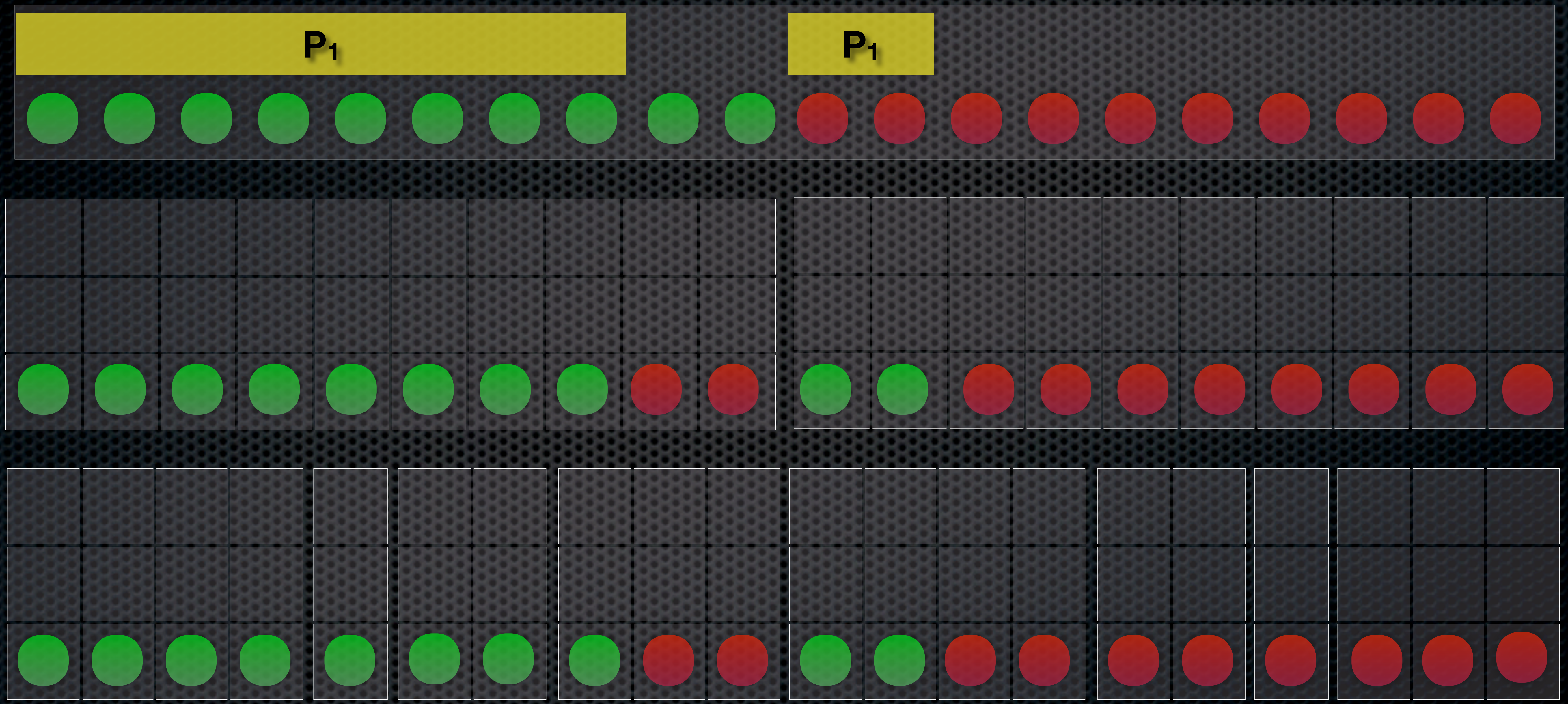
Measure: Information Gain
Optimization: Locally (from DTM)
Partition refinement: Globally (from fCork), explicit (from DTM)

Mine pattern maximizing InfoGain
Partition **all** data using **all** patterns so far
Reiterate on subsets

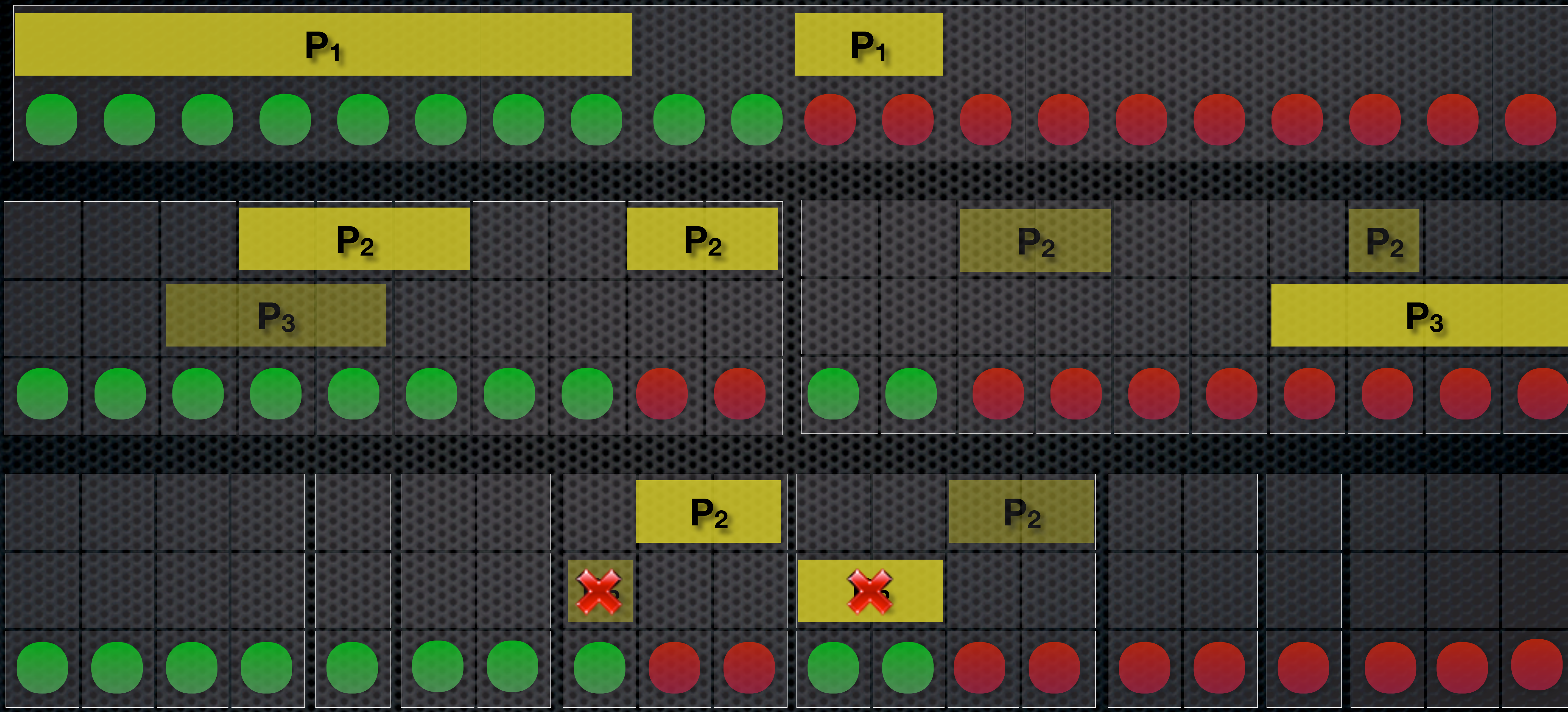
How does it look?



How does it look?



How does it look?



Pros and Cons

Global partition refinement

- More reliable pattern evaluation
- Fewer patterns than DTM, less redundancy

Quickly small subsets

- Faster than either DTM or fCork

Reuses data unless purified

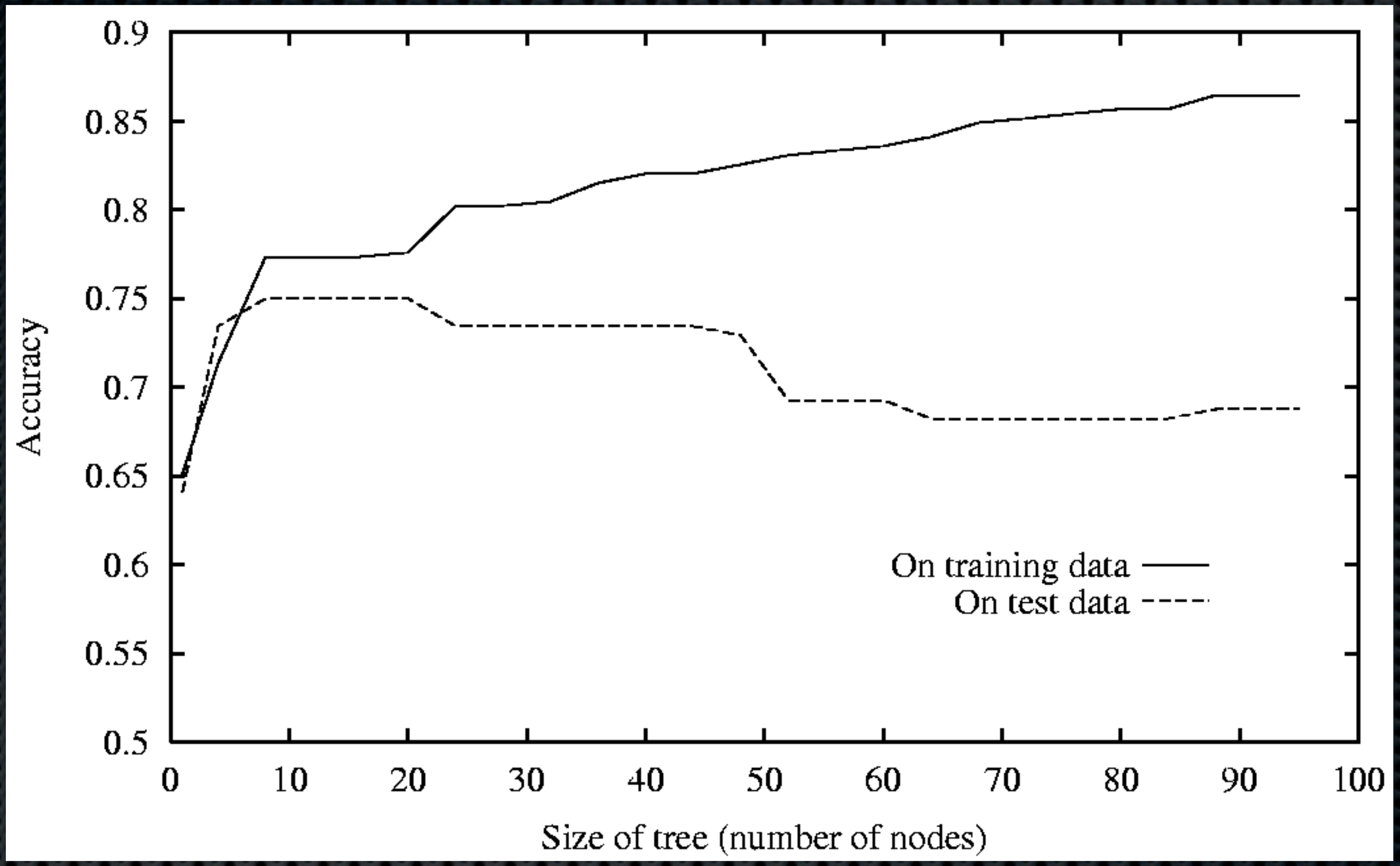
Local measure optimization

Over-fitting *seems* to occur

Loses *some* parallelization capability

- Due to waiting period for all patterns per level

One slide w.r.t over-fitting



Goal **is** effective set w.r.t. target

- I.e. good classification behavior

Fine-tuning patterns to split small subsets can capture noise

- DTM more redundancy, more features, slightly better AUC

Another slide about feature selection

Remember:

- Alleviating the effect of the curse of dimensionality
- Enhancing generalization capability
- Speeding up learning process
- Improving model interpretability

From Wikipedia's entry on "feature selection"

Discussed techniques analogous to subset selection

- Known problem of over-fitting, sophisticated alternatives
- (Cross-)validation possible solution

Others exist

- Feature ranking (top-*k* mining - earlier work)

Between "wrapper" and "filter"
Forward selection

Thank you for your attention

Questions?