Fault Isolation Through the Semi-Supervised Learning of Spatial Patterns in Semiconductor Manufacturing

Nelson D'Amour & Eugene Tuv

Intel/LTD

Talk Outline

Introduction

- Background
- Why SSA (spatial signature analysis) is important
- Challenges and HVM fabs needs

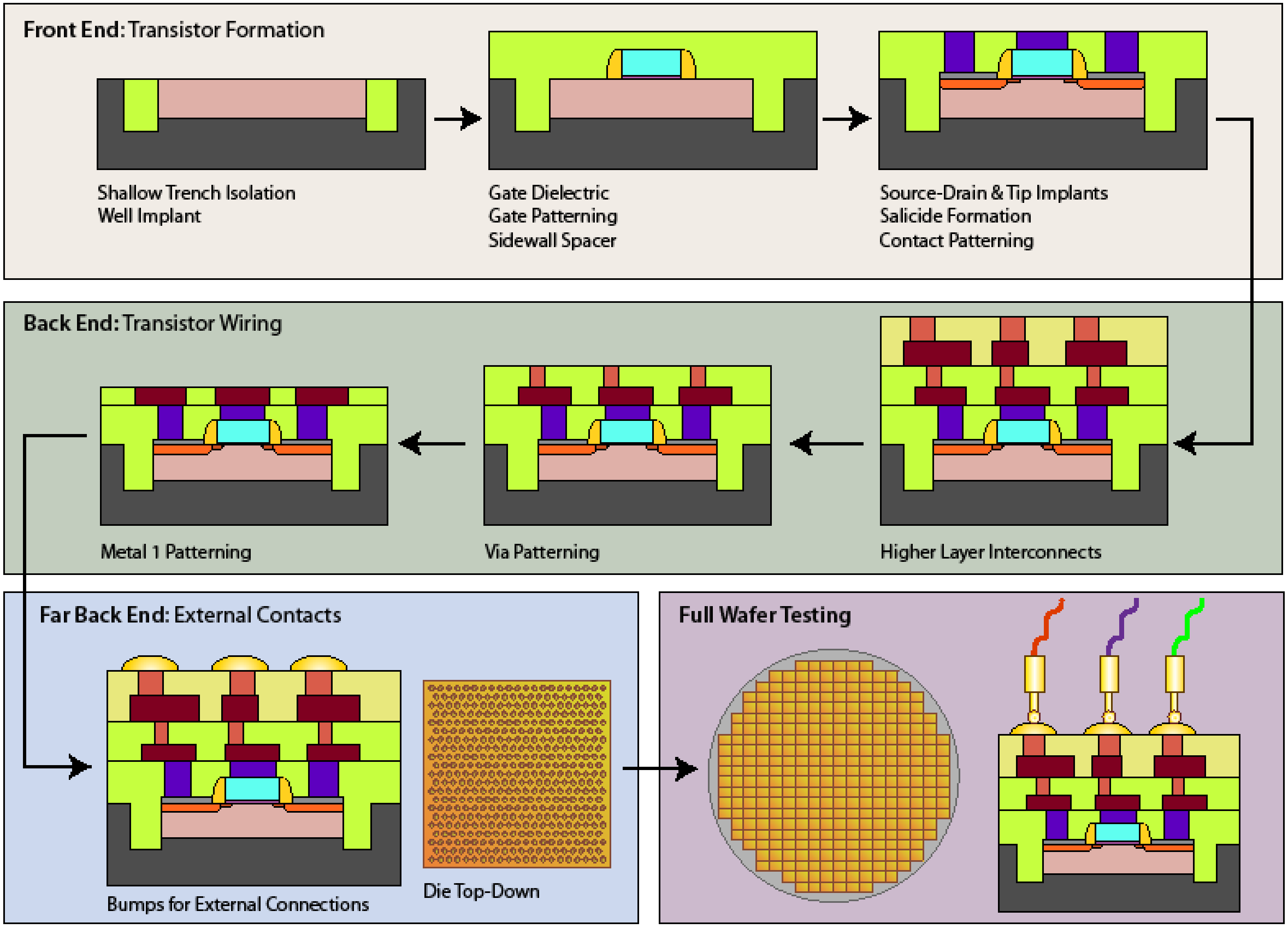
Methodology

- Fault Detection: SSA framework
  - Signature detection
  - Signatures summarization
- Fault Isolation/Commonality Analysis - explaining SS
  - SS Signal enhancement/purification, signature matching
  - Feature Selection. Targeted Rule Induction. Resulting rules point to manufacturing attributes (possible interactions) and time intervals that are likely responsible for SS

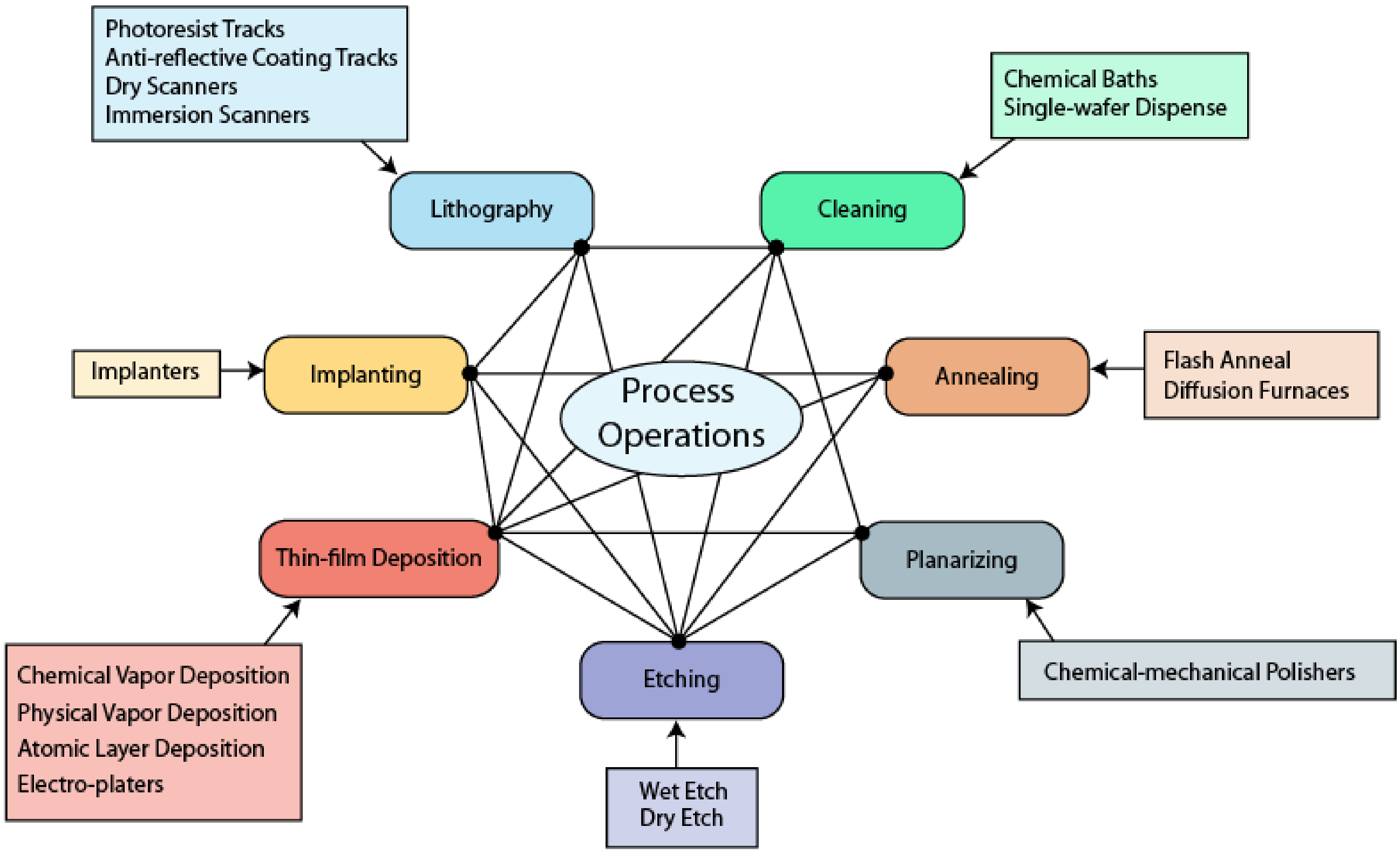
Illustration/Demo

Q&A

Overview of Semiconductor Manufacturing Process



Process Operations to Tools Relationships



Data Collection Scheme

Process History

Process step (102391)
Track-in time (02/22/09 10:29:34.759)
Track-out time (02/22/09 11:05:21.839)
Queue time (384.812 s)
Process time (1875.543 s)
Process recipe (Oxide_Dep_2837_200s_revD)
Tool name (DEP872)
Chamber name (CHA)
Slot position (14)

Analytical Measurements

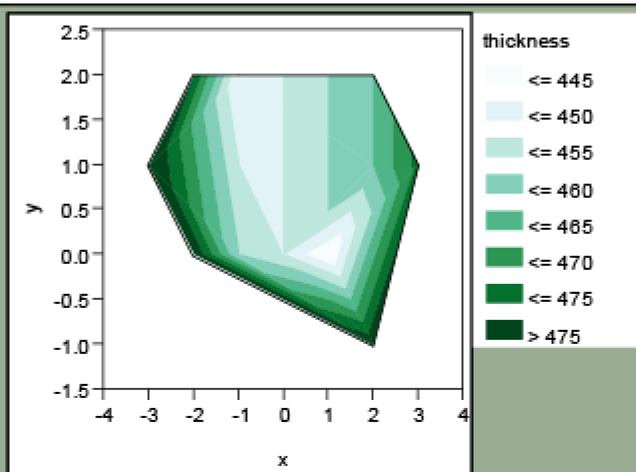
Film Thickness (spatial)
Litho CD (spatial)
Litho Registration (spatial)
Trench Depth (spatial)
Etch Rate (point)
Polish Rate (point)
Defect Particles (spatial)

Tool Processing Parameters

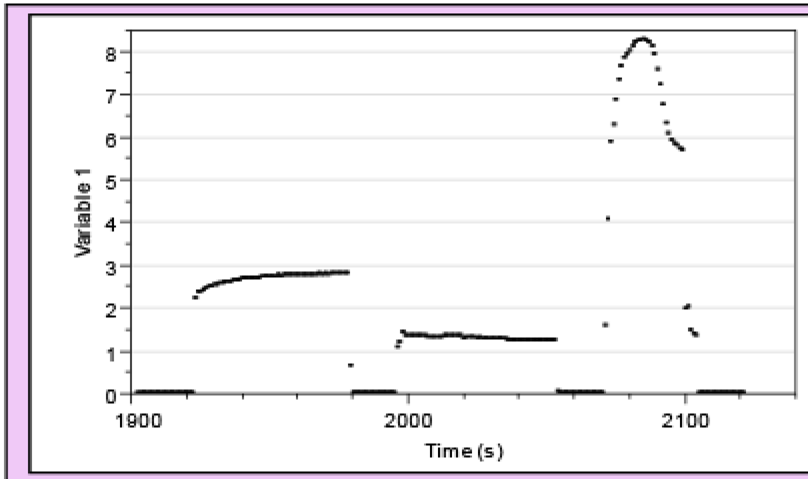
Pressure (time series)
Temperature (time series)
RF Power (time series)
Bias Voltage (time series)
Gas Flow Rate (time series)
Chemical Flow Rate (time series)
Bath Life (time series)
Plasma Discharge Counter

Electrical Testing

Functional
Design Structures
Cache Patterns
Parametric Data



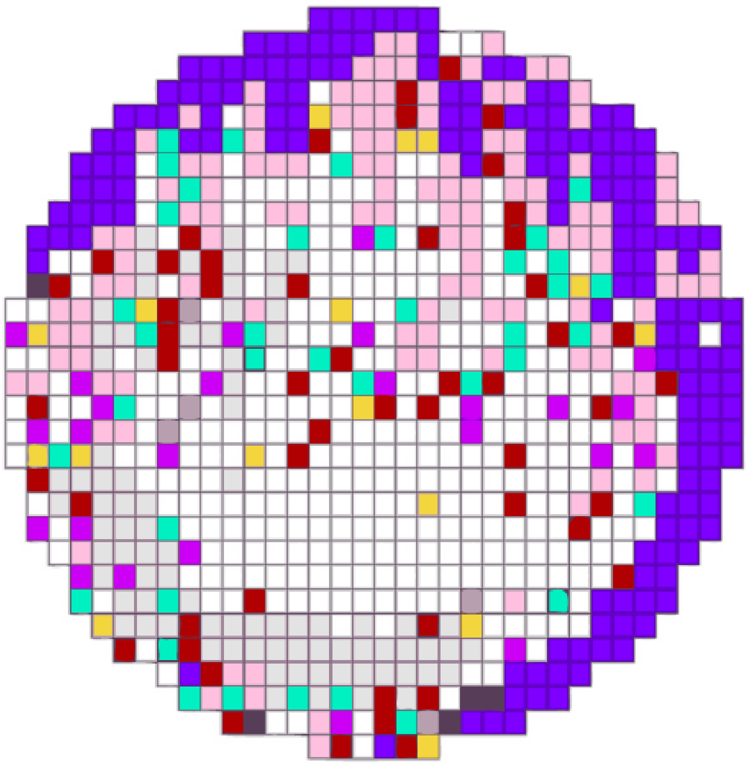
Sample Film Thickness Contour Map



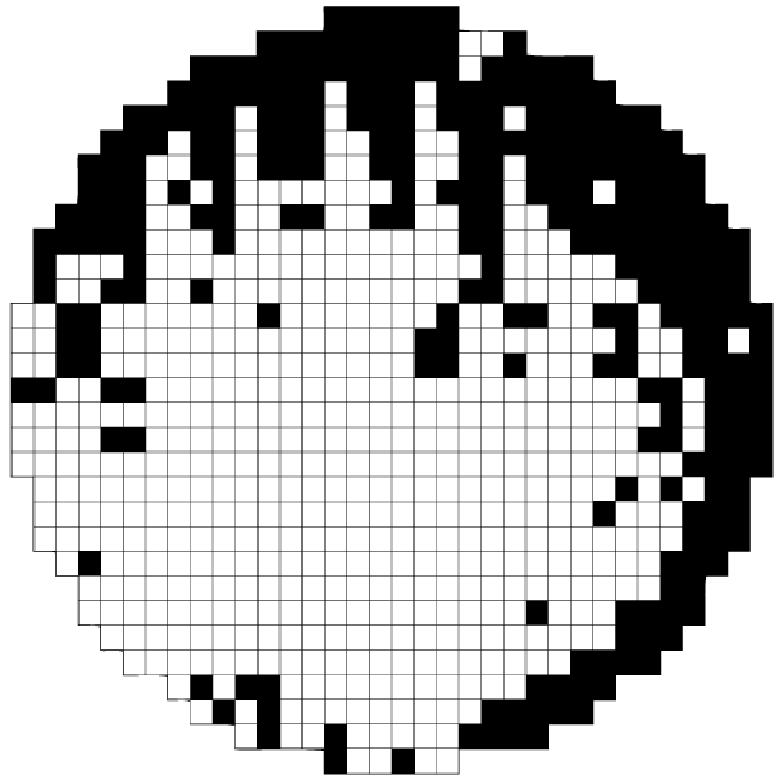
Sample Tool Variable Trace

Spatial Map of Electrical Testing

- One example of testing before the wafer is sawed and packaged into parts is Bin testing.
- Since there are many types of bins that are similar, the first step is to group them into the target that one wants to do commonality/correlation to.



Full Bin Map



Binary Target Map

Problem statement / challenges

Problem: Majority of fab yield problems have a spatial signature.

Ultimate goal: isolate in time manufacturing steps/tools/toolattributes responsible for the excursion that manifested itself as non-random spatial failure signature.

Challenges:

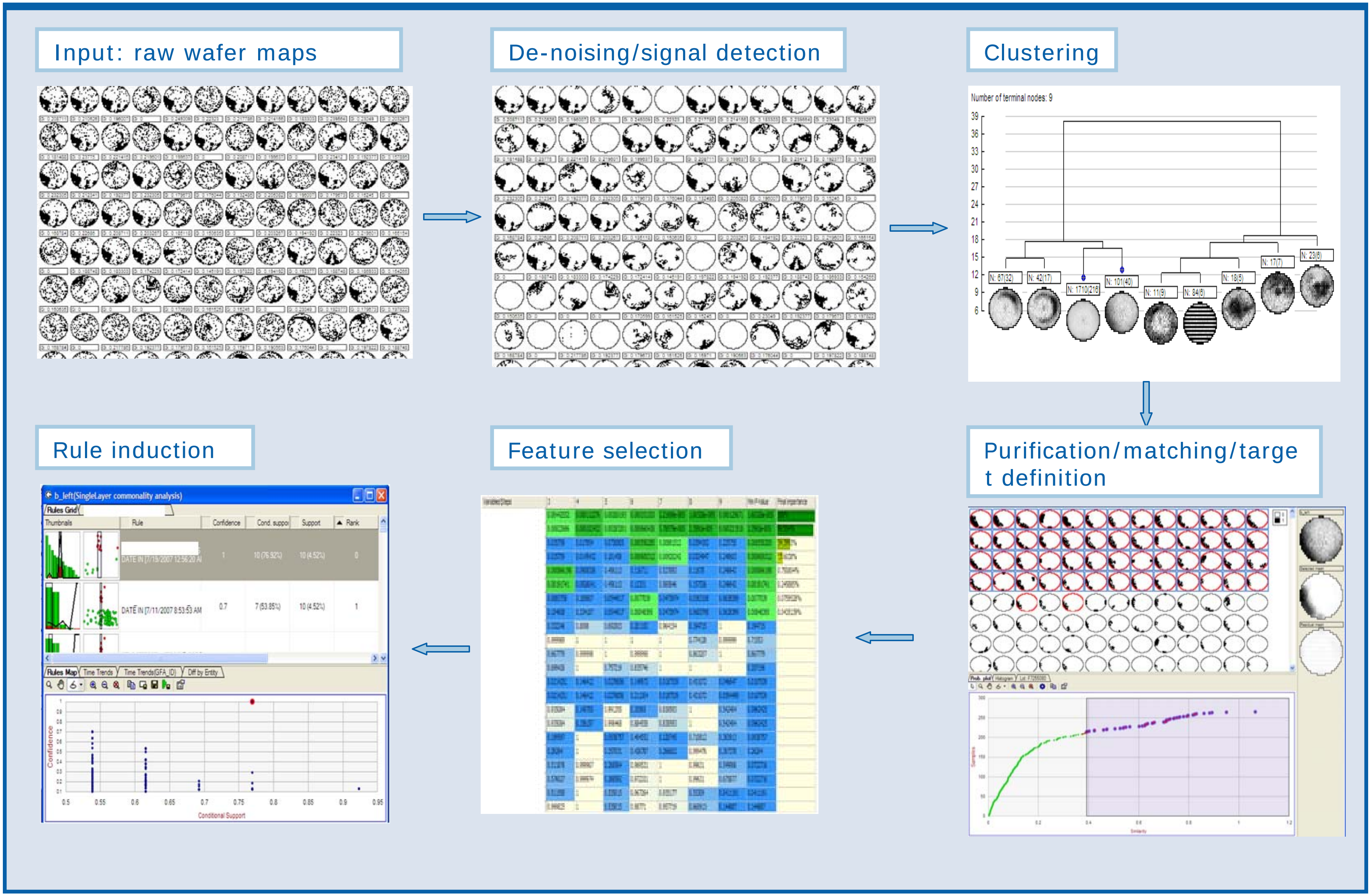
Spatial Signature analysis:

- Detect non-random spatial signatures on wafer maps for all relevant levels of aggregation (lots/lot sets, wafers/wafer sets, ...) dynamically ("on the go") for potentially thousands of wafers in minutes with standard hardware
- Integrate automated groupings/classification of spatial signatures with interactive improvement by engineers
- Purify/match spatial signatures → create a target variable for the supervised commonality learning (distance to a signature and/or signature ID)

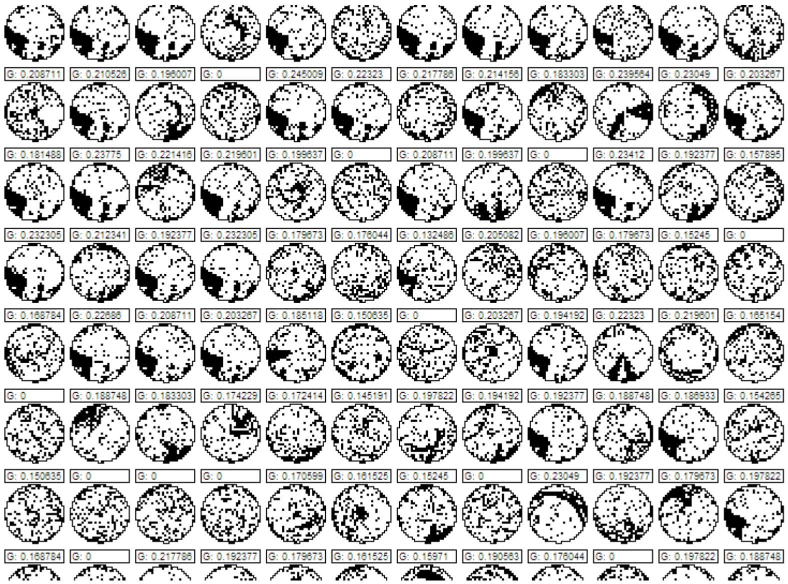
<u>Commonality analysis:</u>

- Find operations and tools (by time) that explain purified spatial signatures (➔ next: ID tool EP params ➔ root cause diagnosis)

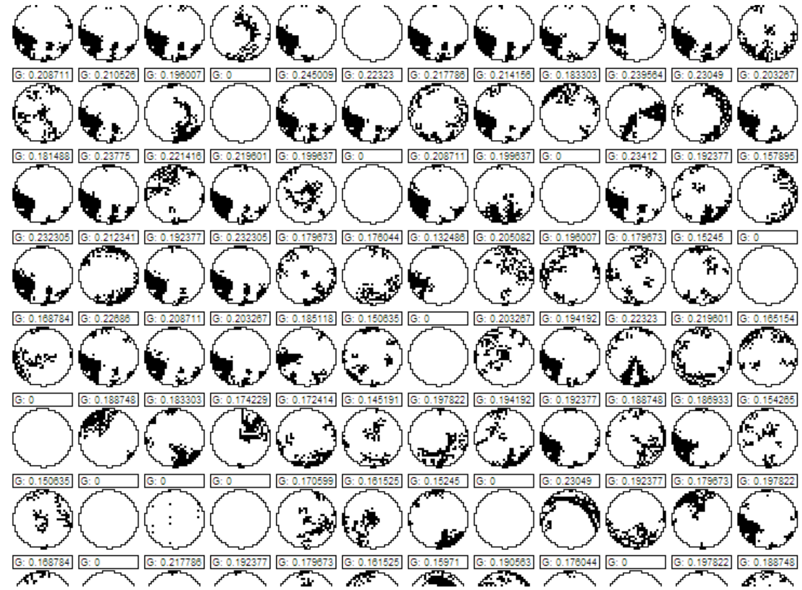
Basic Analysis Flow



SSA Elements: non-random spatial signatures



Raw wafer maps



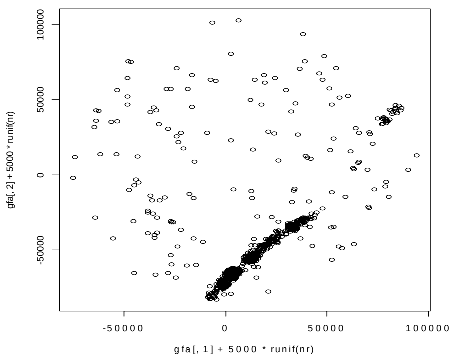
De-noised wafer maps

- We use a de-noising procedure to separate spatial signatures from the random distribution and assign a *non-randomness* index to each wafer.
- Wafers with the same total failure rate could have dramatically different *non-randomness* indexes.
- *Non-randomness* index could be used to separate/rank wafers with SS and/or in APC settings

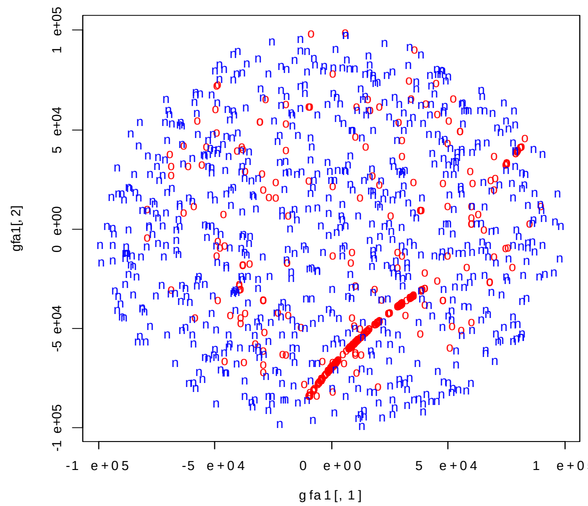
De-noising approach using supervised learning

- Build a committee of experts (classifiers) to detect regions of non-random failure densities.
- Each expert is trained to distinguish a given wafer signature from a *random signature* of the same cardinality

De-noising approach

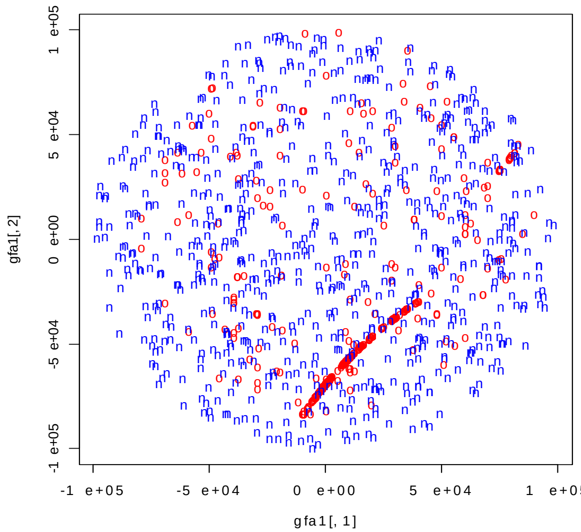


Original defect signature

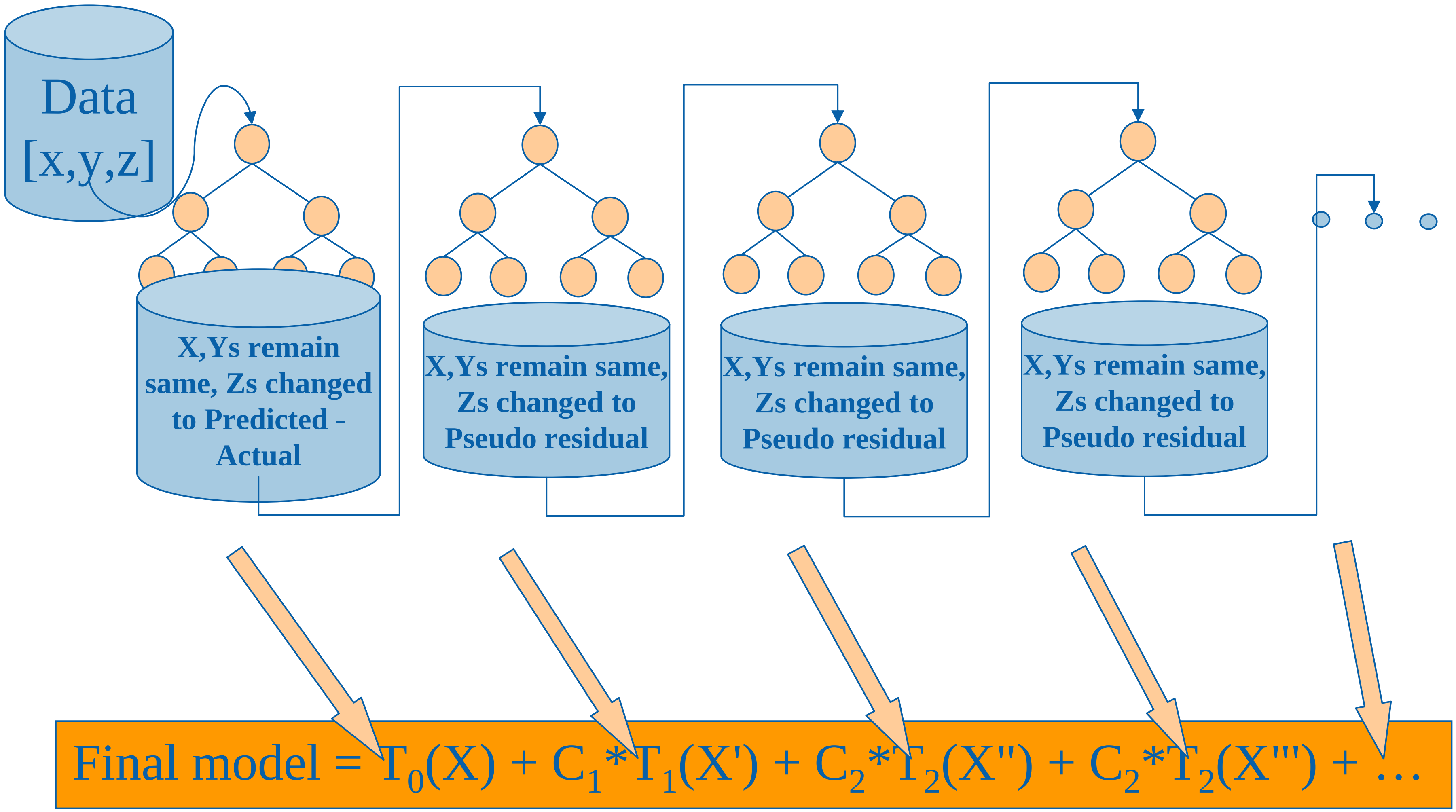


Each expert gets a different, independently constructed, uniformly distributed over the wafer *random signature (blue)* + *original signature (red)*

Training data has 3 vars (x, y,z) : x,y coordinates and a binary target z(“red”, “blue”).



Classifiers used: gradient tree boosting machine



De-noising approach

- Each failure point **x0** on the wafer gets a point estimate of probability pi(**x0**) that it belongs to a *random signature* from each of the experts of the committee.
- All these estimates represent IID sample by construction of size = #experts
- Now we can use a standard statistical inference to call a point on a wafer “random”

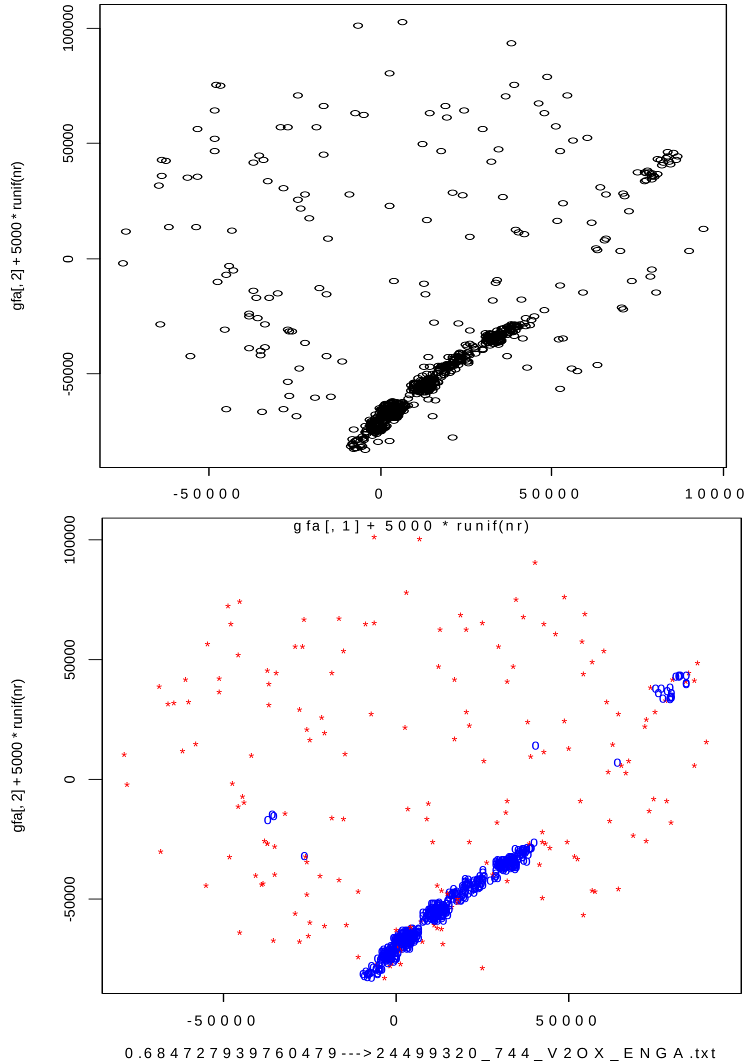
De-noising approach

We used the following criteria: *upper 99% conf limit on p_hat(**x0**) < 0.5*

Example:

Upper graph represents a wafer pattern before the de-noising

Red points are identified as a “random dust” – to be swept, blue ones are part of “non random” signature – kept, used to calculate a non-randomness index



De-noising approach

That approach works well in off-line mode or when the #wafers to be processed is not very large (hundreds)

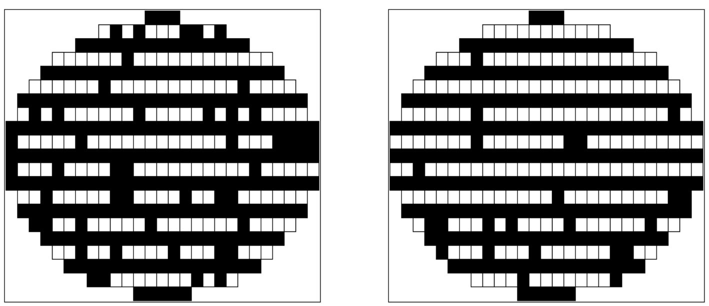
For an interactive mode with 1000s of wafers we offer a computational shortcut to denoising

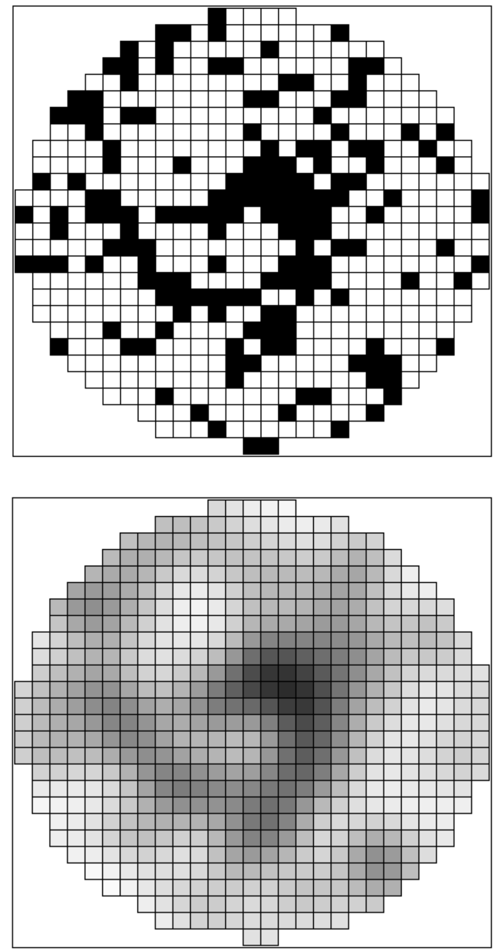
- Standardized density in each point of a smoothed wafer is compared to an 1-p quantile threshold estimated through interpolation from a number of random wafers generated for a range of different average defect densities

SSA elements: binary (0/1) assignment at every die position is true with some uncertainty

Spatial kernel based smoothing with optional edge effect weighting is applied

Special algorithms/tests are used for periodic signatures





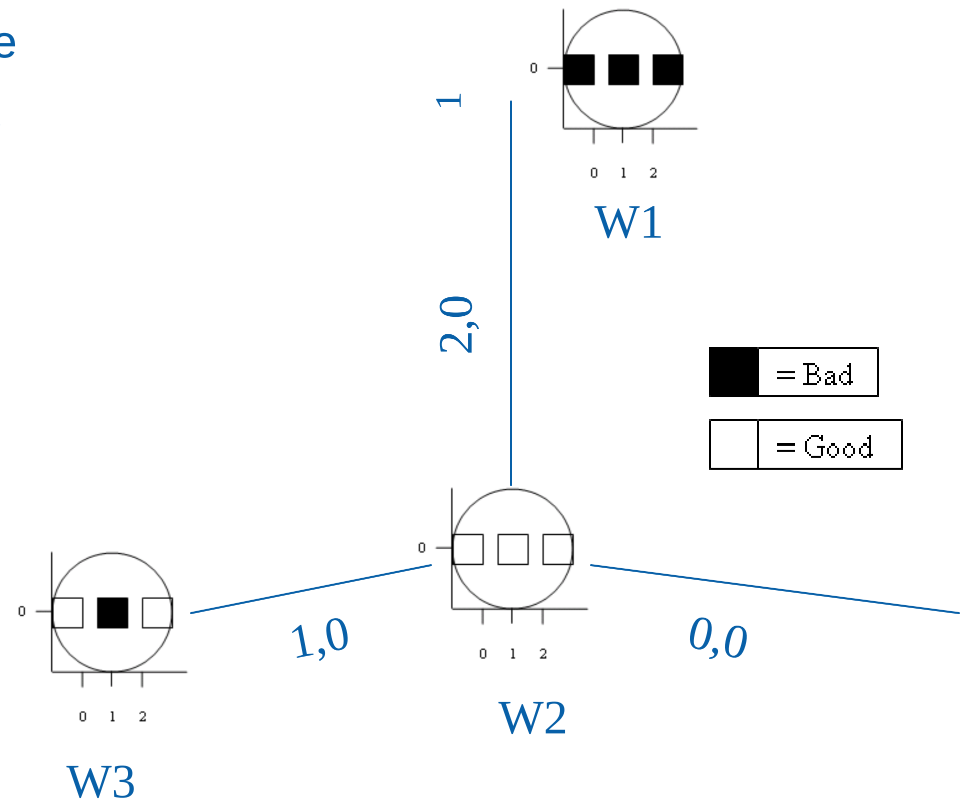
SSA elements: Signature Summarization/Clustering Basic Concept

Each die location is an axis in space

For example, 3 wafers, each with 3 die locations, bin result is either "good" (bin = 0) or "bad" (bin=1)

Wafer	die location (x,y)	Bin
1	0,0	1
1	1,0	1
1	2,0	1
2	0,0	0
2	1,0	0
2	2,0	0
3	0,0	0
3	1,0	1
3	2,0	0

Example wafer maps



SSA elements: Signature Summarization/Clustering wafer level clustering

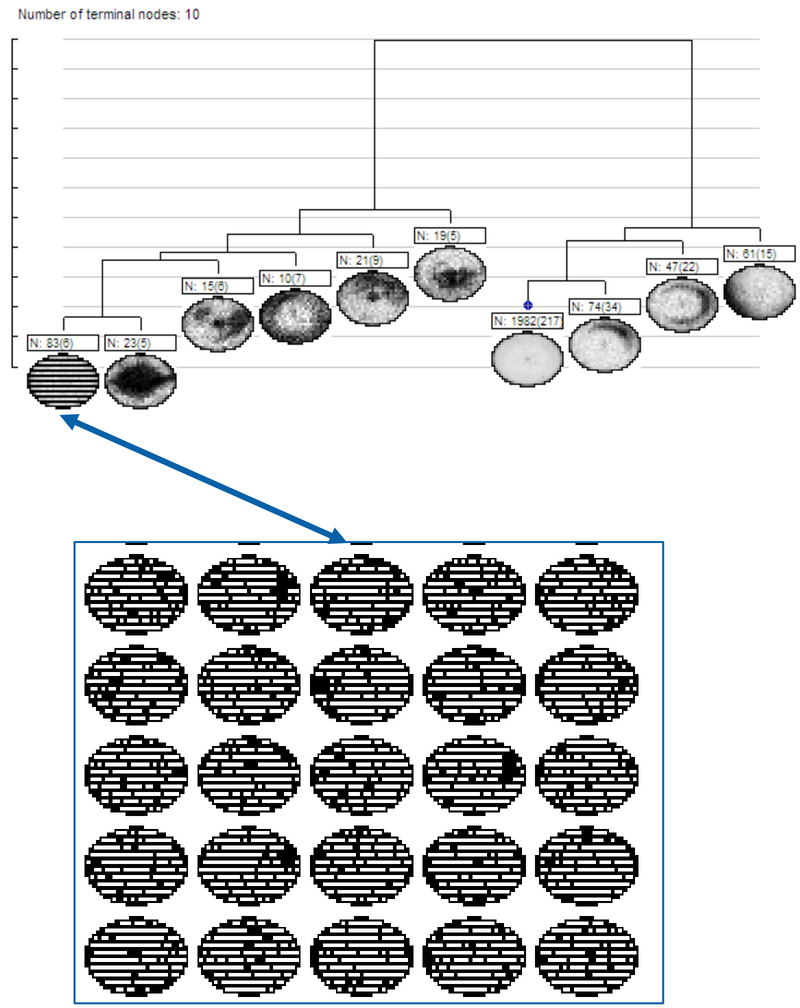
Clustering is done on smoothed and exponentially transformed (d->d^p) wafers after no-signature and "dog" wafers were removed

hybrid divisive-agglomerative clustering

special initialization with a large number of distinct seeds (far from each other)

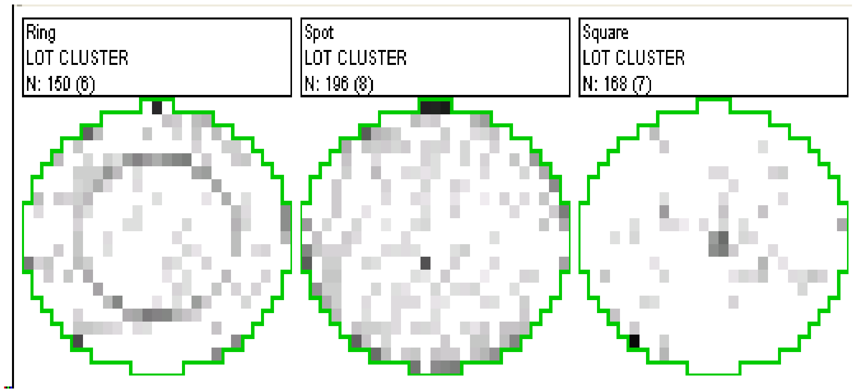
k-means like update to the seeds with outlier-cluster removal

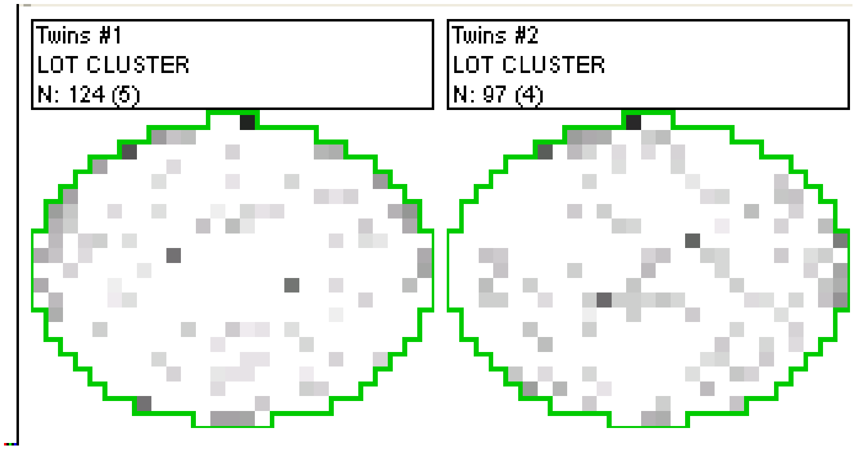
then agglomerative clustering is applied resulting in interactive expandable drillable dendodram-tree



SSA elements: Signature Summarization/Clustering cross wafer/lot small signatures

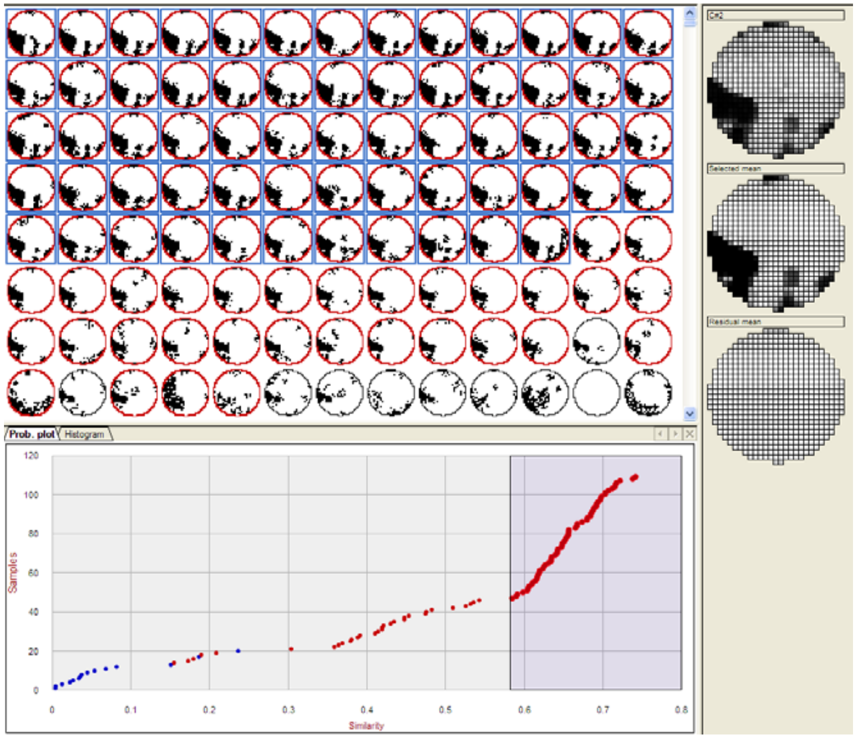
- Small cross wafer/lot signatures invisible on the wafer level, almost always missed by humans
- Requires completely different algorithm
- Uses a set of thresholds and a special similarity measure described in matching





SS Signal enhancement/purification signature matching

- Crucial for the commonality analysis (ultimate goal) ability to purify the identified groups (each having pure signature).
- To do this, a similarity score (distance) is required which accurately measures how well a specific wafer matches a pattern (preferably close to a human eye).
- It is reduced to supp(A1&A2)-K*supp(A1\A2+A2\A1) if both searched signature and smoothed wafers have common areas A1,A2 of the same density.
- Affected wafer/lot indicator (0/1) and distance to a pattern become the targets for the commonality analysis.



Commonality Analysis - explaining Spatial Signatures.

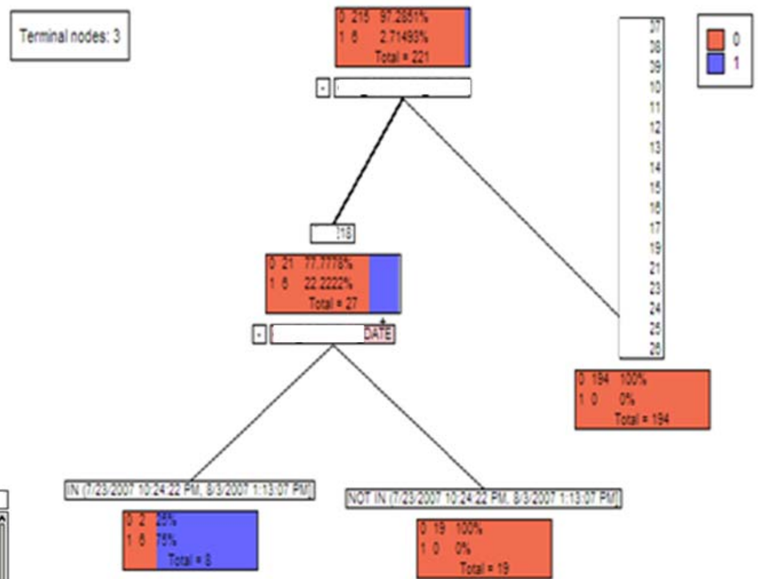
Feature selection: removing irrelevant suspects-variables

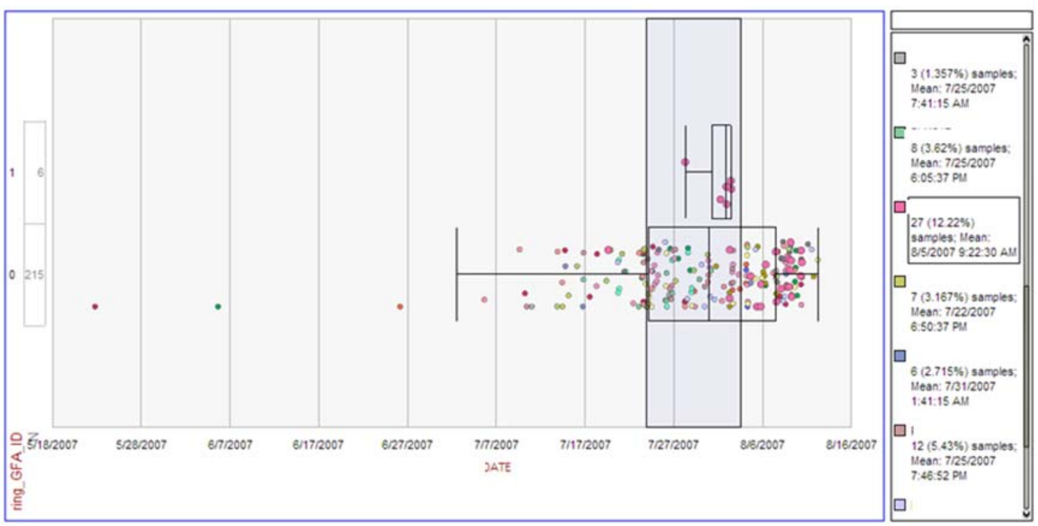
- Signature Indicator (or a distance to a pattern) becomes a binary target, process attributes – predictors
- Random Forest (RF) is a committee of experts (trees) learner, and is capable of efficiently ranking features for very large datasets.
- We exploit this property of RF, augment the original data with artificial probe variables constructed independently from the target, and use their ranking for removal of irrelevant variables from the original set.
- Selected subset of relevant variables will be used in rule induction to discriminate wafers/lots that have a signature from the rest of the population

Variables\Steps	3	4	5	6	7	8	9	Min P-Value	Final importance
3044_	0.00442532	0.000112276	0.00200193	0.000331333	2.21698e-005	1.60328e-005	0.000125671	1.60328e-005	100%
3044_	0.00822686	0.000102432	0.00283201	0.000660438	5.78579e-005	2.3592e-005	0.000221518	2.3592e-005	96.5537%
2014_	0.035759	0.017854	0.0730805	0.000550285	0.00881512	0.0294192	0.225755	0.000550285	34.3993%
2014_	0.035759	0.0145432	0.181458	0.000600312	0.00920242	0.0324947	0.248603	0.000600312	15.6038%
1867_	0.000866198	0.0908336	0.456113	0.116711	0.527893	0.11678	0.248642	0.000866198	0.7508%
1867_	0.00191741	0.0926041	0.456113	0.12331	0.565846	0.157326	0.248642	0.00191741	0.245885%
3589_	0.0883758	0.185607	0.0544017	0.0077039	0.0473974	0.0363108	0.0638399	0.0077039	0.0759526%
3589_	0.104608	0.334187	0.0544017	0.00848395	0.0473974	0.0683795	0.0638399	0.00848395	0.0435139%
356_	0.332246	0.8098	0.692003	0.201182	0.964154	0.194715	1	0.194715	
308_	0.999989	1	1	1	1	0.774138	0.999999	0.71553	
346_	0.667779	0.999998	1	0.999998	1	0.863207	1	0.667779	
378_	0.699426	1	0.757219	0.835746	1	1	1	0.207156	
396_	0.0214251	0.146412	0.0276856	0.149972	0.0187539	0.431072	0.046647	0.0187539	
396_	0.0214251	0.146412	0.0276856	0.211304	0.0187539	0.431072	0.0594499	0.0187539	
415_	0.935084	0.140705	0.991355	0.30565	0.838593	1	0.542404	0.0962425	
415_	0.935084	0.156157	0.998468	0.884558	0.838593	1	0.542404	0.0962425	
418_	0.199597	1	0.0938757	0.404552	0.120745	0.710812	0.363913	0.0938757	
418_	0.26264	1	0.357031	0.426787	0.266602	0.999476	0.367378	0.26264	
436_	0.511876	0.999967	0.268564	0.969531	1	0.99631	0.549906	0.0722716	
436_	0.576027	0.999974	0.268592	0.972331	1	0.99631	0.675877	0.0722716	
446_	0.511558	1	0.535015	0.967264	0.935177	0.55309	0.0411191	0.0411191	
446_	0.999825	1	0.535015	0.98771	0.957719	0.660913	0.144887	0.144887	
447_	0.281279	0.400233	0.592109	0.109597	0.363459	0.999877	0.128138	0.109597	

Commonality Analysis. Decision Tree (DT) is one of the most basic and widely used rule induction engines

- Variables selected by the FS become input variables to the DT construction algorithm
- Our tree is capable of an interval search, and a look-ahead for linked/nested variables such as tool-ID and time-through-tool





Commonality Analysis. We employ a custom direct search rule induction engine

- We employ custom rule induction engine to search for Tool x Time dependencies covering maximum # of affected wafers/lots with high confidence.
- (**C**onfidence, **C**onditional**S**upport) plane, each point is a rule
- Rules are found to maximize a goal function - weighted sum of **C** and **CS**

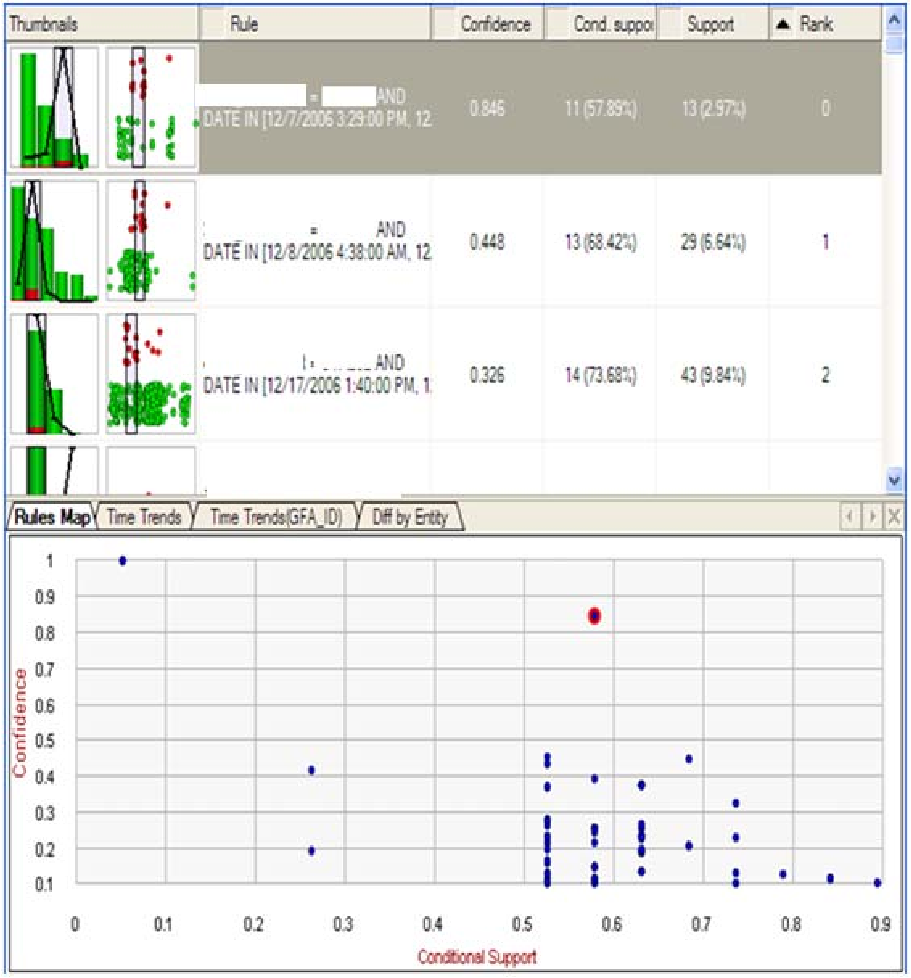
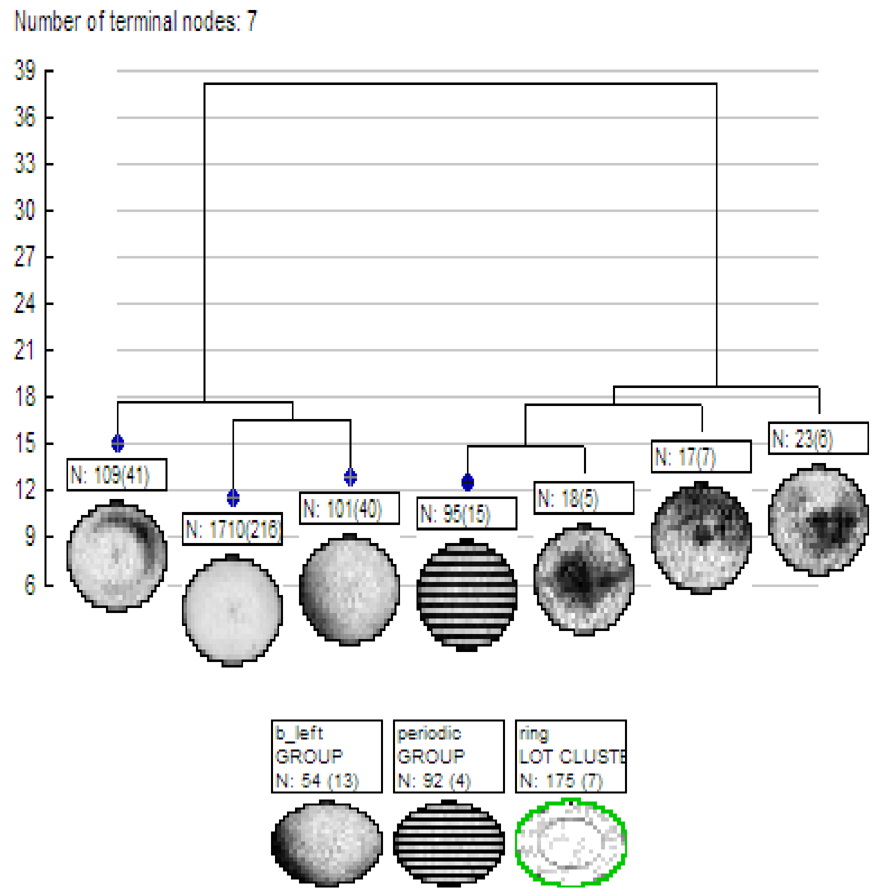
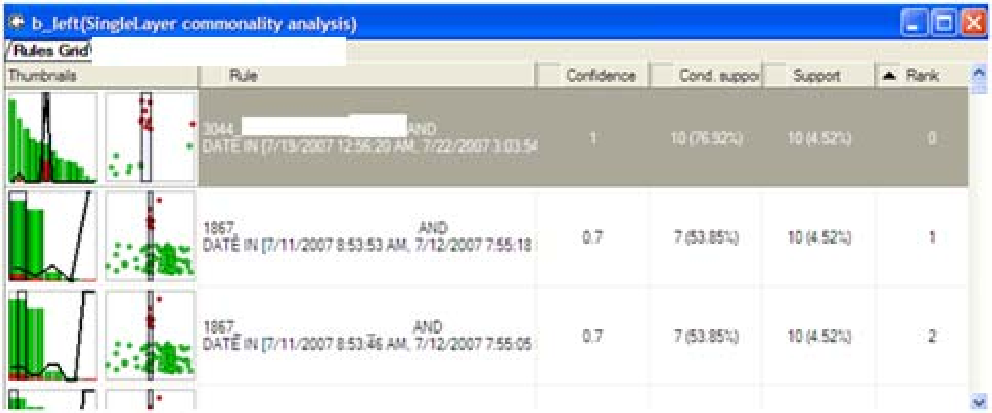


Illustration / Demo

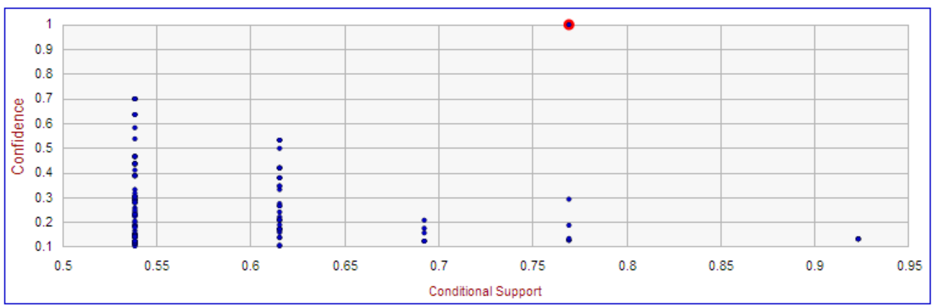
- Data set contains almost 5,000 wafers
- Signatures processing and clustering the wafers, we arrive at the groupings shown to the right
- There are several interesting spatial patterns in this views, will focus on the highlighted pattern
- Cluster data is combined with fab tool data, and rule induction is used on purified signature indicators

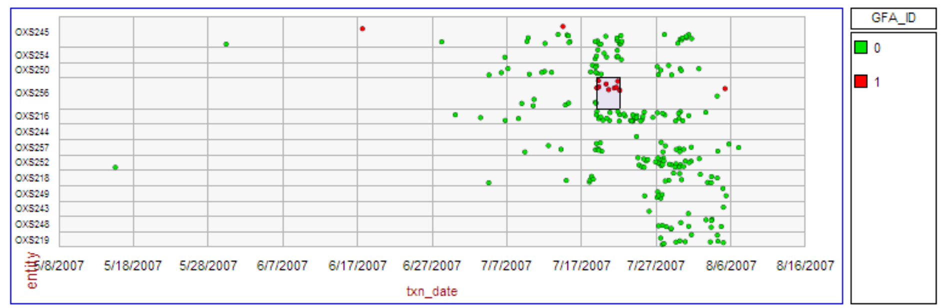


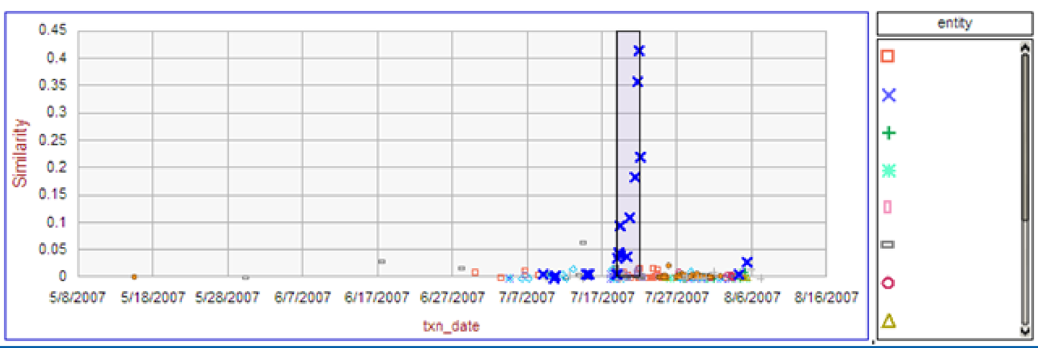
Bottom left signature



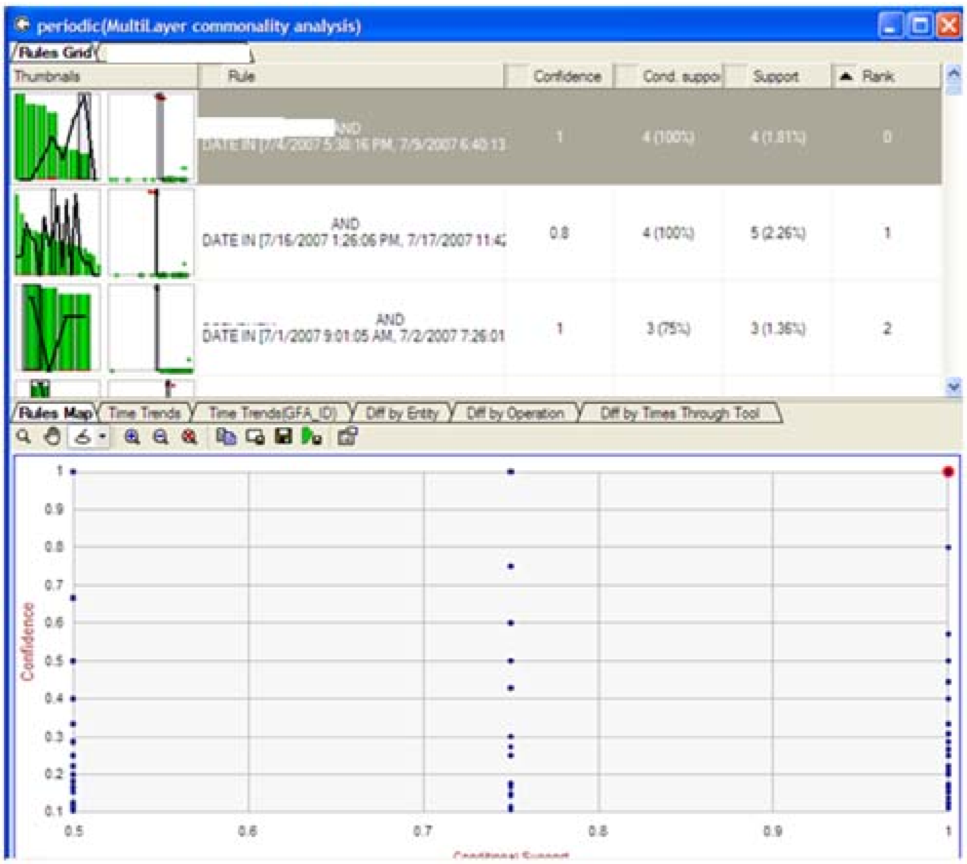
Thumbnails	Rule	Confidence	Cond. suppor	Support	Rank
	3044 AND DATE IN [7/19/2007 12:56:20 AM, 7/22/2007 3:03:54	1	10 (76.92%)	10 (4.52%)	0
	1867 AND DATE IN [7/11/2007 8:53:53 AM, 7/12/2007 7:55:18	0.7	7 (53.85%)	10 (4.52%)	1
	1867 AND DATE IN [7/11/2007 8:53:46 AM, 7/12/2007 7:55:05	0.7	7 (53.85%)	10 (4.52%)	2

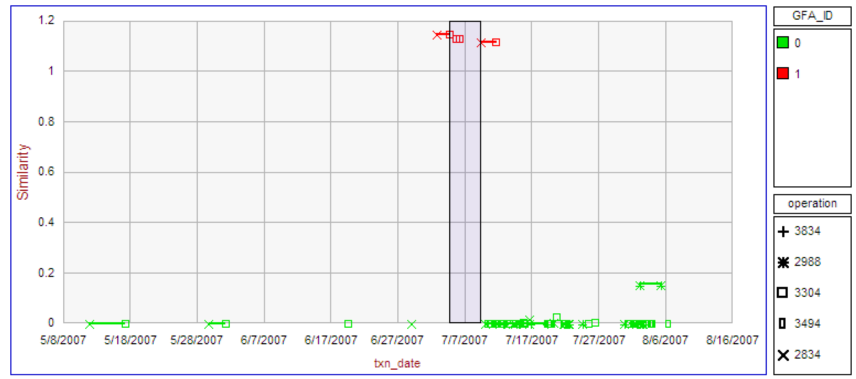


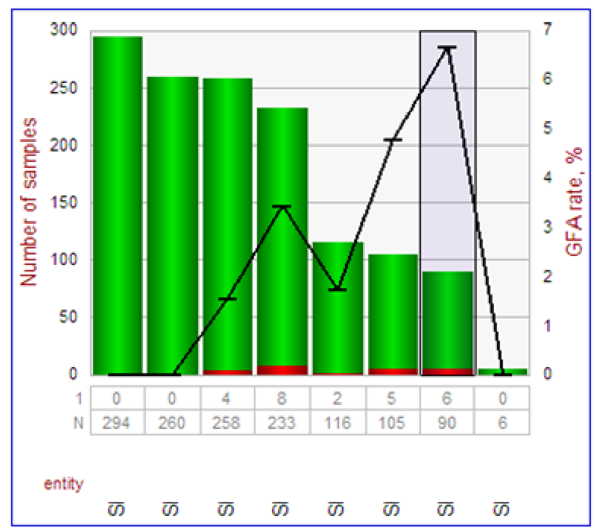




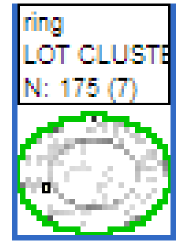
Periodic signature

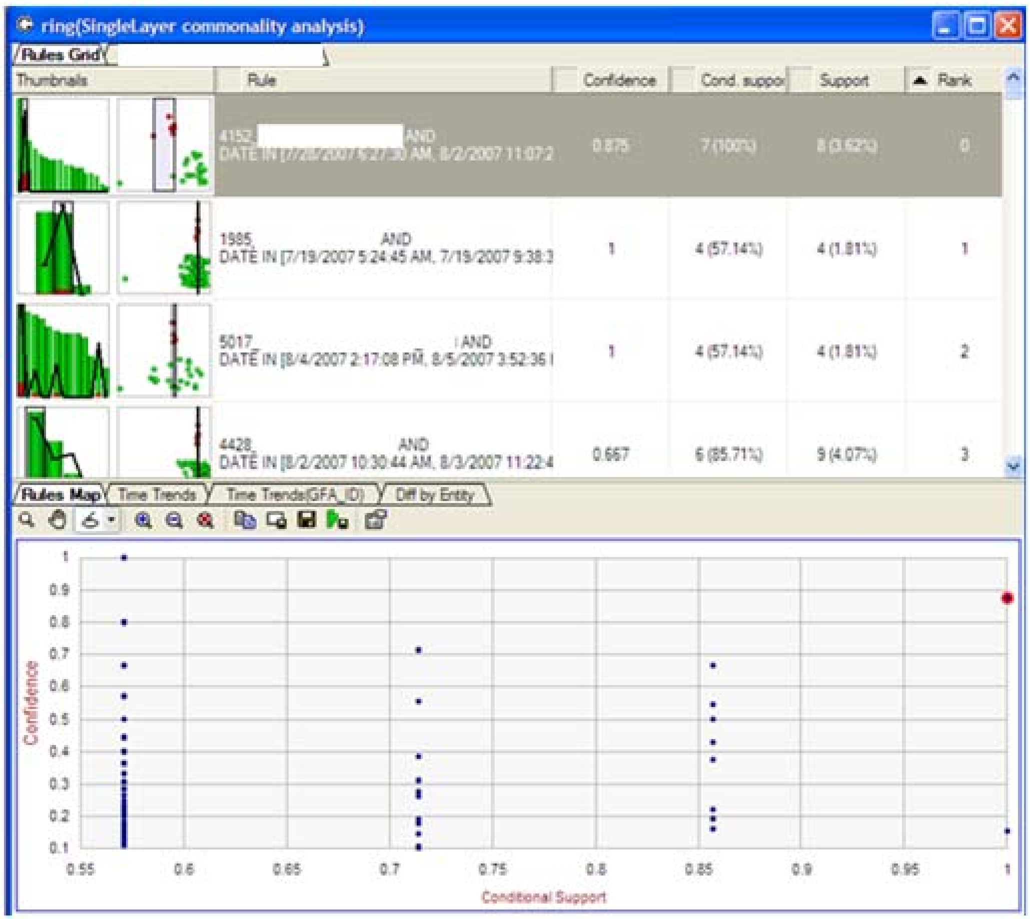


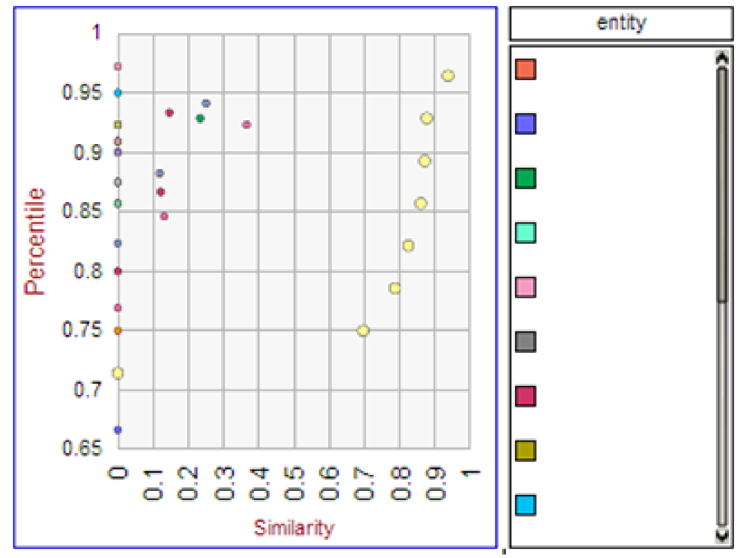


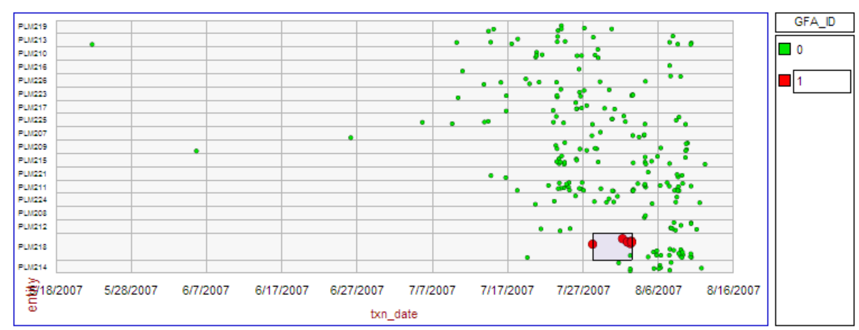


Ring cross-wafer signature









Conclusion

- To do this entire analysis has required about 5 minutes; a great improvement over the approximately 4-10 hours required by highly skilled analysts (often with no success) with the set of techniques previously used.
- We have applied these methods to many data sets where the spatial patterns and their commonalities were known, were able to identify them all correctly, and find quite a few that were previously undetected.