Image processing and simple classification for filtering candidate exoplanets

(Hadrien* Cambazard, Nicolas* Catusse, Antoine Chomez, Anne-Marie Lagrange)

*G-SCOP
Operations Research / Combinatorial Optimization

Intro and overview - key ideas

Question: Separate signal from noise to identify exoplanets

PACO [Flasseur et al 2018]:

Produce a probability distribution that can be interpreted as an SNR map

- Detection is performed a statistical test
- Avoid self-subtraction of signal
- Too many artifacts can be left and the candidates need to be filtered
- Filter based on SNR threshold

SNR threshold → **Prediction threshold**

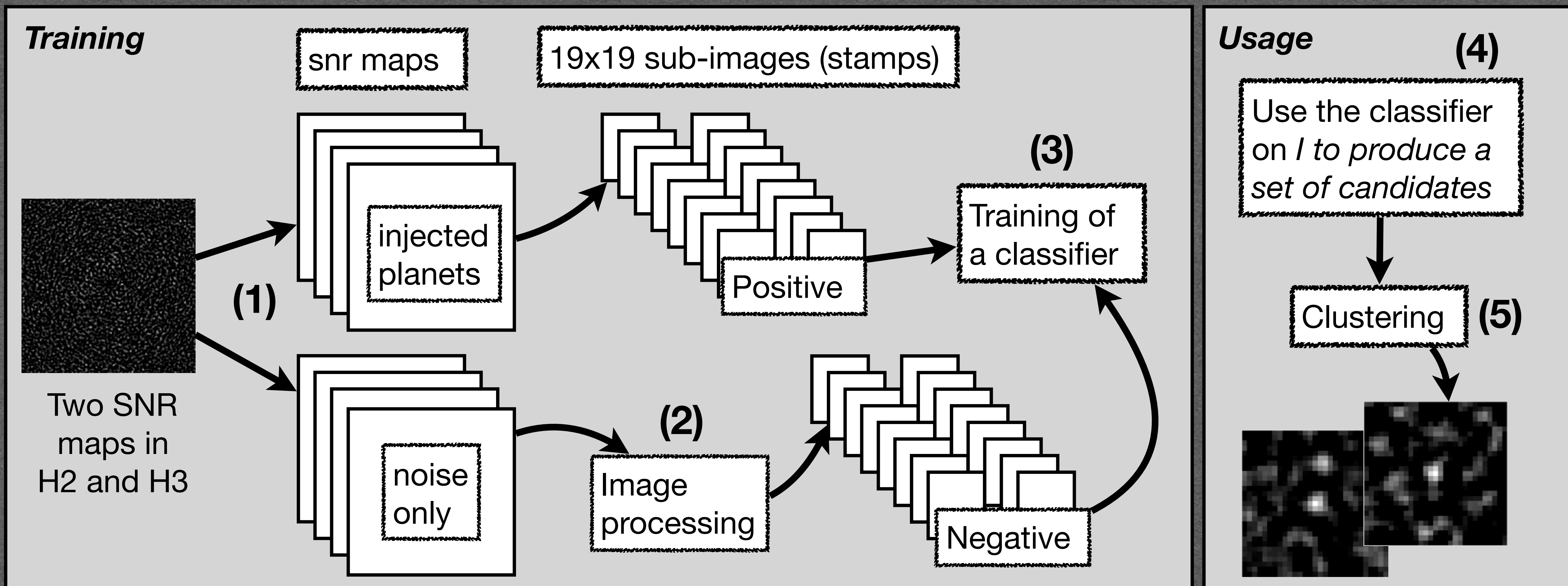
Threshold

Exoplanets missed

False detections

- Keep the statistical approach of PACO which is very powerful
- Incorporate additional physical features into the threshold

Intro and overview - key ideas



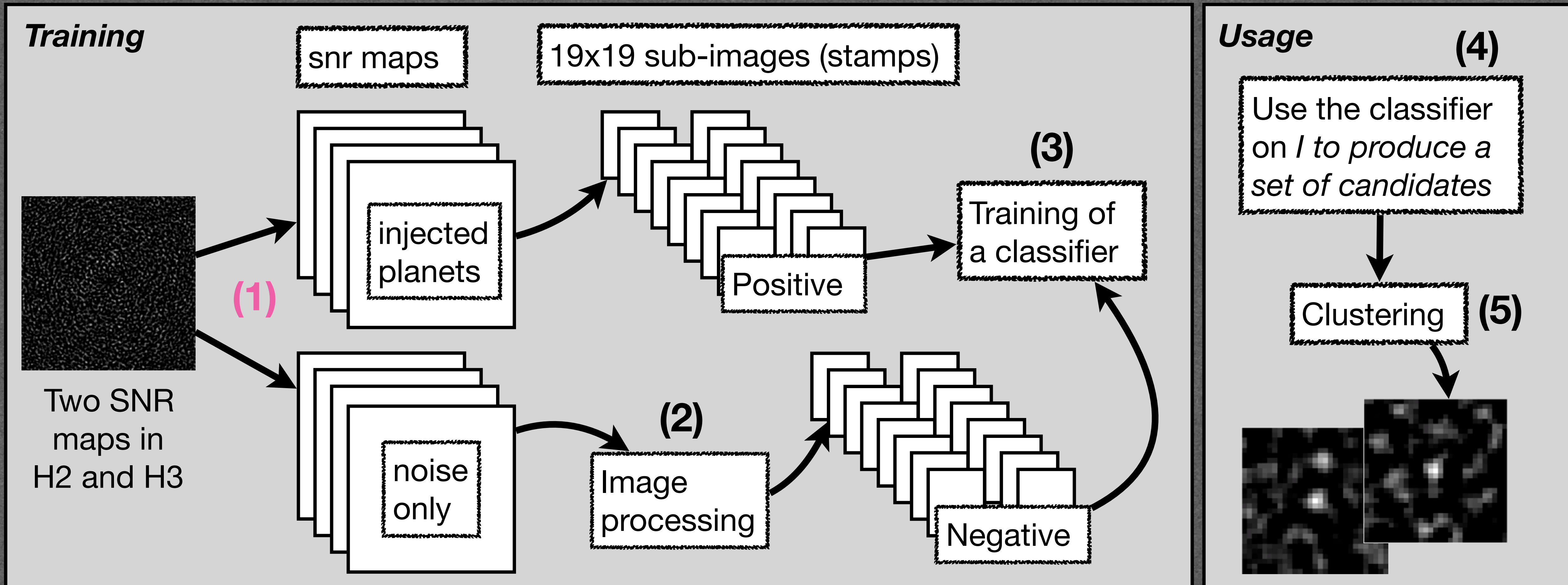
Overall process

- Input: Two SNR maps (H2 and H3) produced by PACO from a single 4d data-cube
- Approach based on **supervised learning** using **sub-images:**

  Stamp / sub-image : 19 x 19 pixels

  Stamps with **known** class (positive = contains a planet, negative = noise/speckle…)

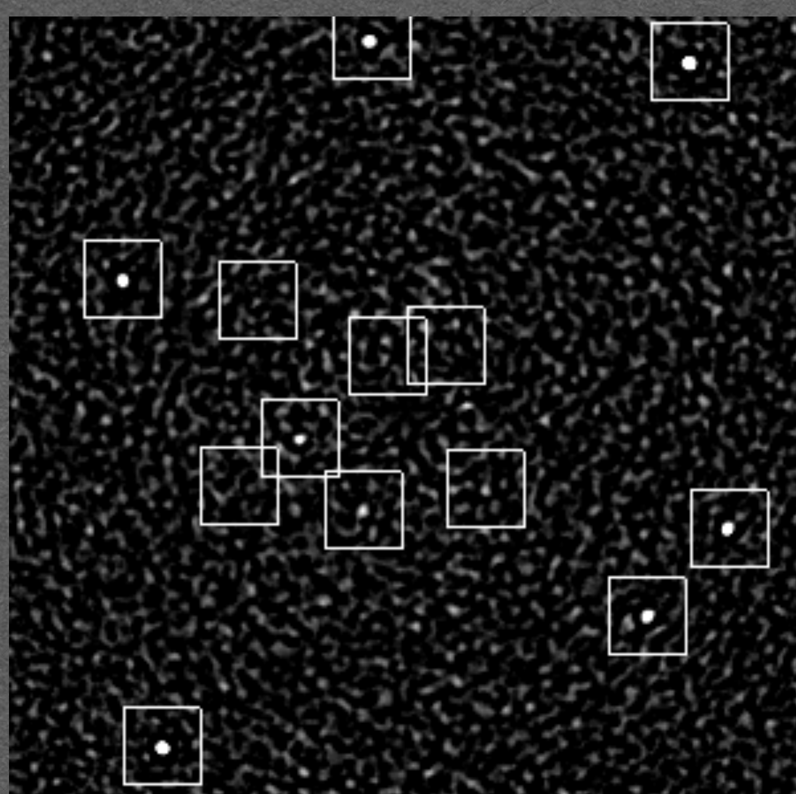
- A classifier **is built from** a given SNR map and its usage is **dedicated** to this map



(1)

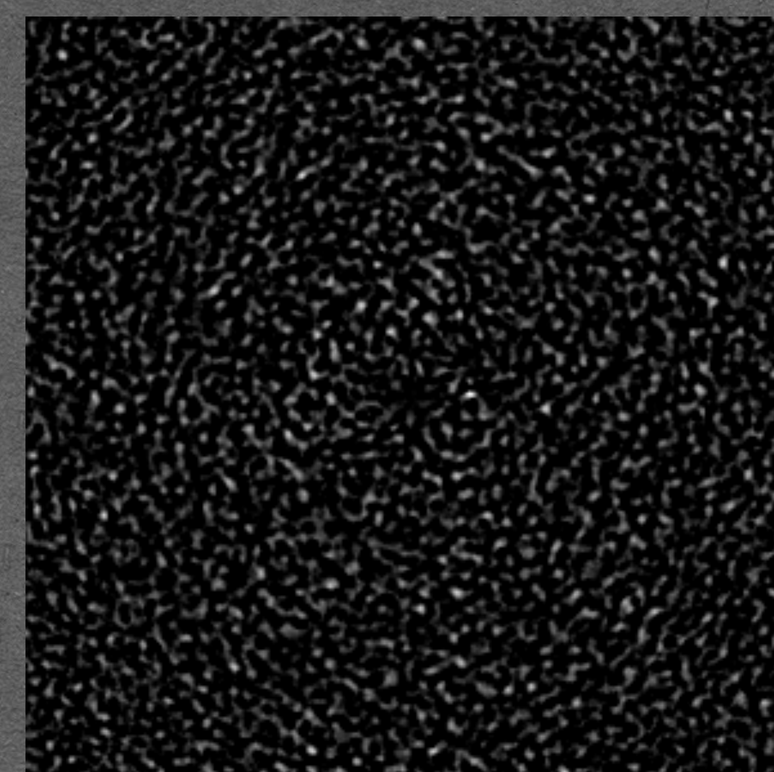
injected planets

noise only

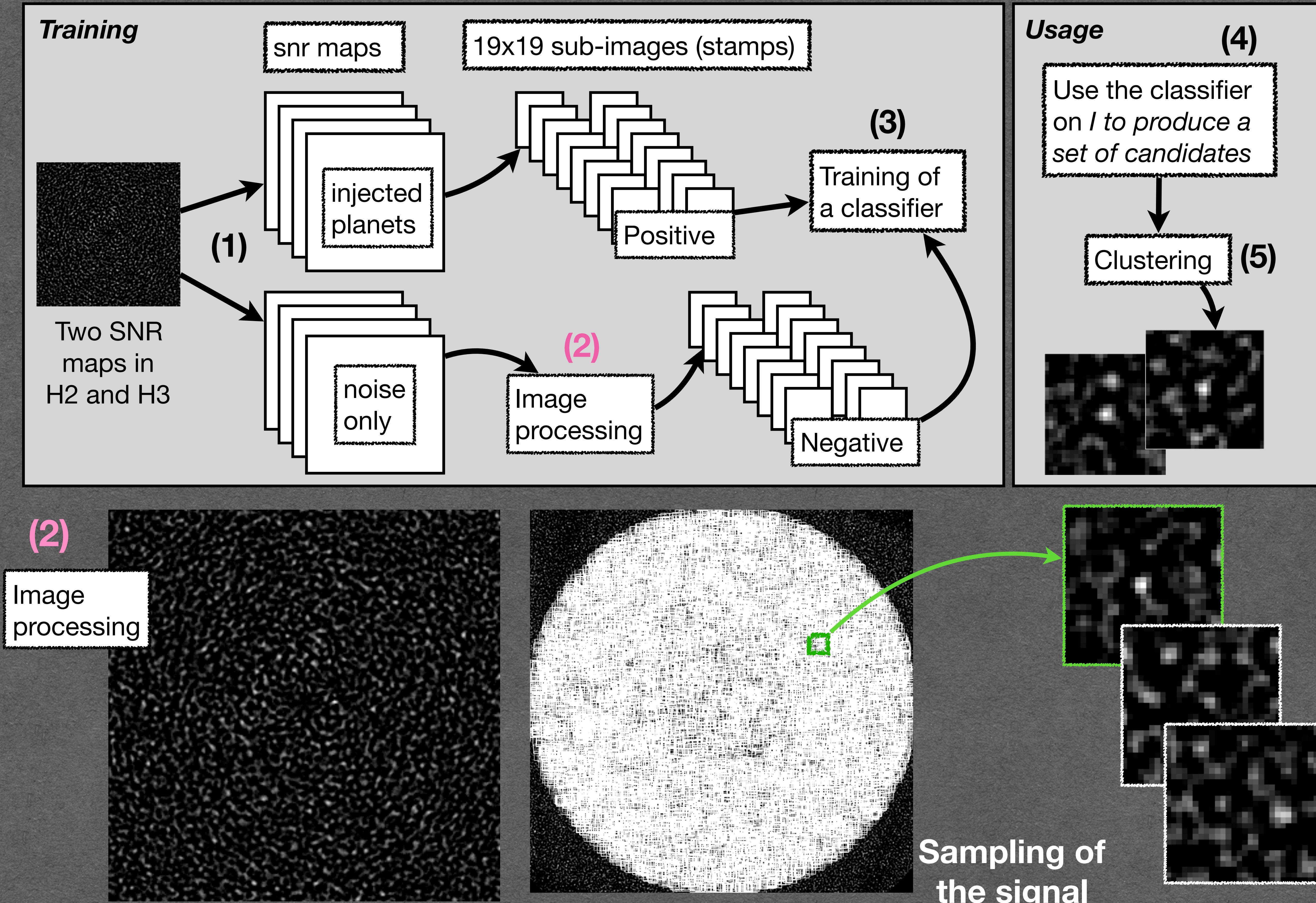


Positive samples (injections)

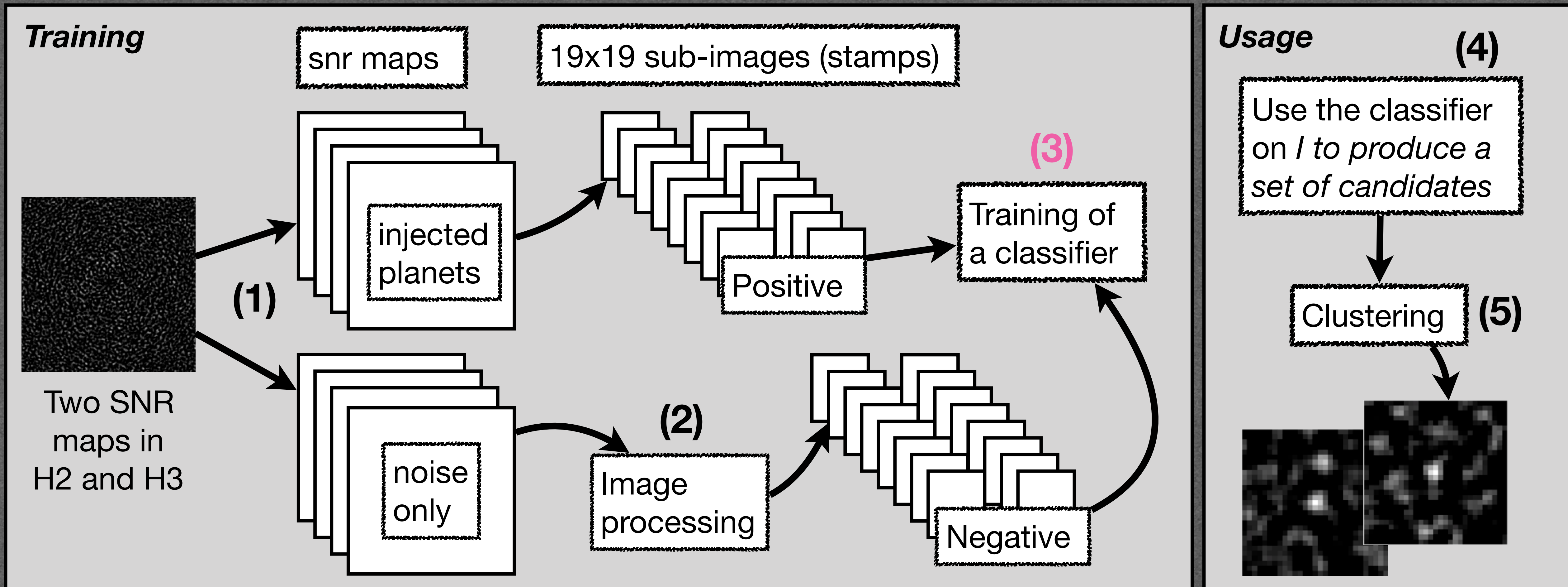
Supervised learning need positive/negative samples



SNR map with guaranteed noise (cube with inverted rotation)



- Edge detection algorithm to target peaks of SNR
- Generate a sub-image centered around each SNR peak



(3)

Training of a classifier

Simple features related to meaningful notions:

- Snr values in H2 and H3
- Norm of gradient of Snr values
- Airy figure
- Speckle
- … and a number of other attempts

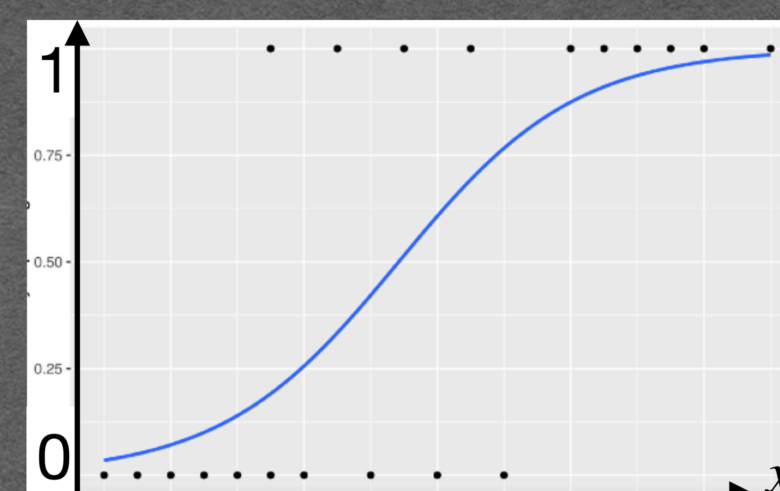
Classification using a **logistic regression** (training with cross-validation)

$$\Theta(x) = \theta_0 + \theta_1 x_1 + \theta_1 x_2 + \ldots + \theta_n x_n$$

$$\frac{1}{1 + e^{-\Theta(x)}}$$

sigmoid function

Probability of a stamp to be an exo-planet



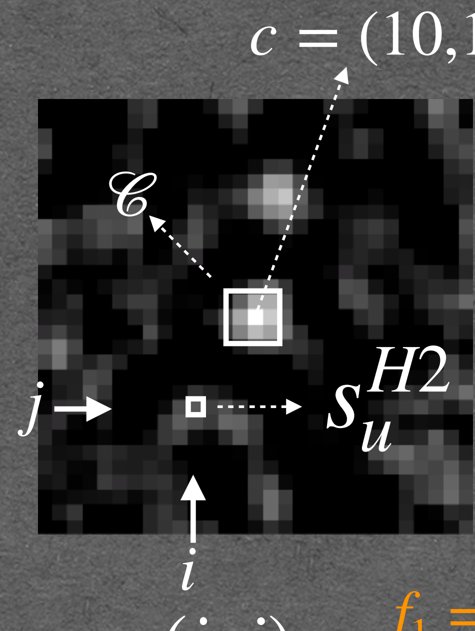
wikipedia

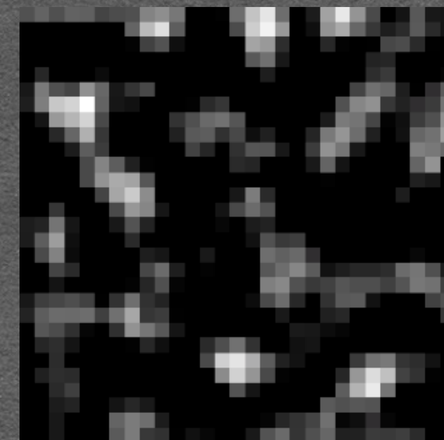
Features

(3) Training of a classifier

A stamp = a 19x19 matrix of pixels

Snr matrix in H2 | Snr matrix in H3

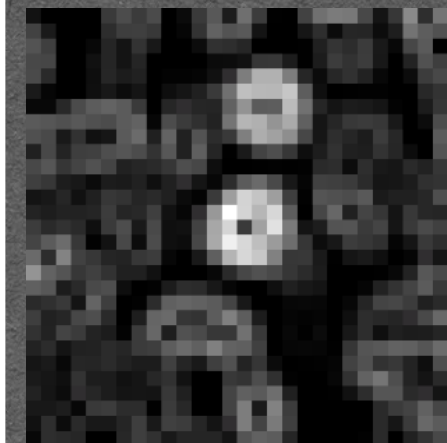




$u = (i, j)$

$f_1 = \max(\{s_u^{H2} \mid u \in S\})$

Norm of gradient in H2

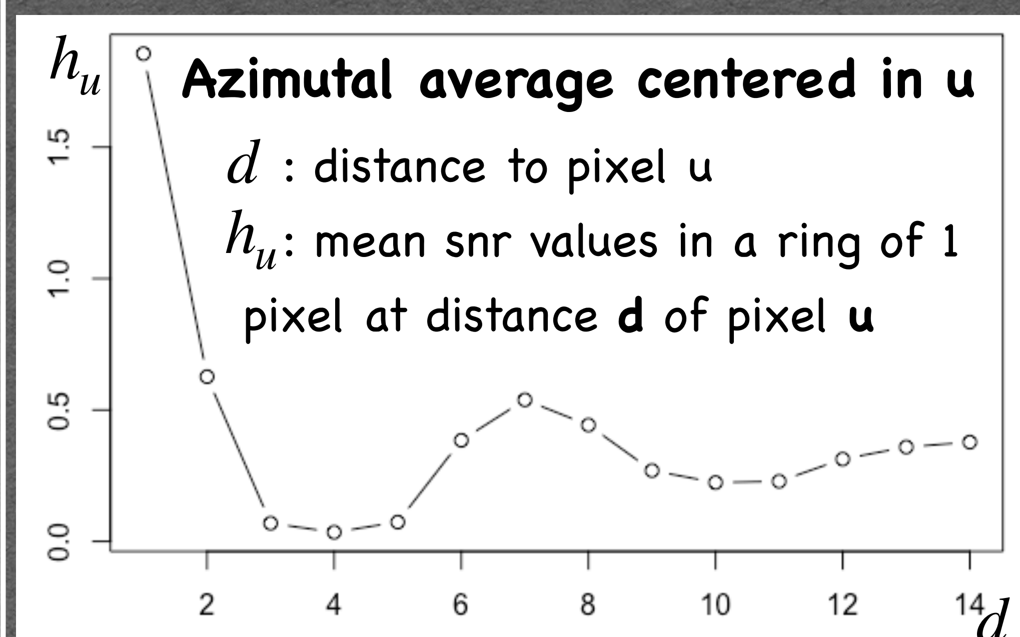


$f_2 = \mu(\{g_u \mid u \in S\})$

$f_3 = \max(\{g_u \mid u \in S\})$

$f_4 = \sigma(\{g_u \mid u \in S\})$

Consequences in the Snr map of the Airy figure of the image

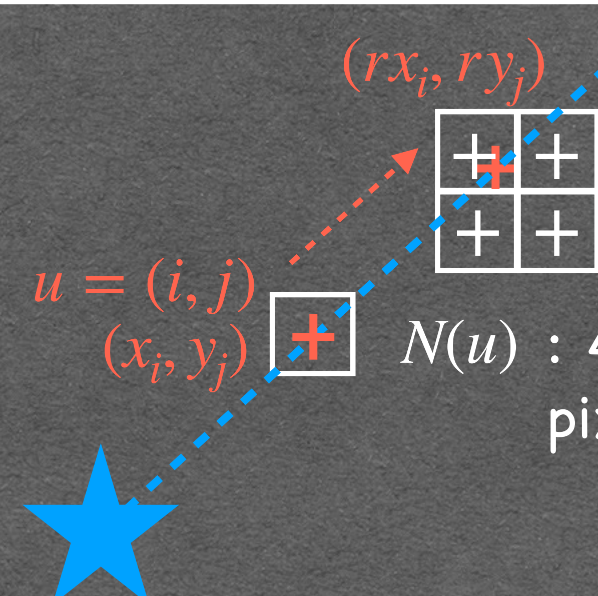


$$\sigma(u) = \sigma(\{h_u(d) \mid d \in [1,14]\})$$

$$f_5 = \max(\{\sigma(u) \mid u \in \mathscr{C}\})$$

Speckle

Attempts to characterize a speckles using chromaticity



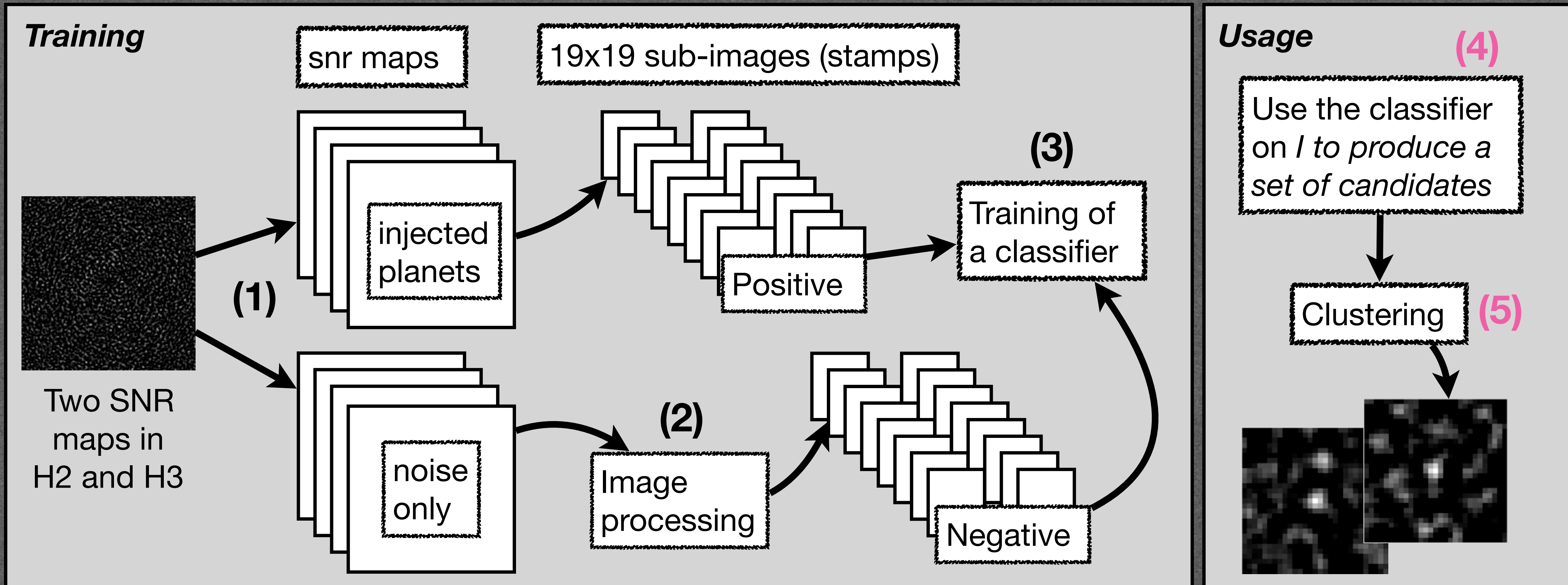
$r = \frac{\lambda_{H2}}{\lambda_{H3}}$

$N(u)$: 4 closest pixels of (rx_i, ry_j)

$w(u)$: estimation of the snr at location (rx_i, ry_j) using the four closest pixels and their distance to (rx_i, ry_j)

$$sp(u) = \frac{s_u^{H2} - w(u)}{\max(s_u^{H2}, w(u))}$$

$$f_6 = \mu(\{sp(u) \mid u \in \mathscr{C}\})$$



(4) Use the classifier on *I to produce a set of candidates*

(5) Clustering

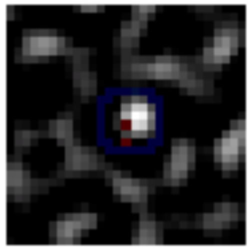
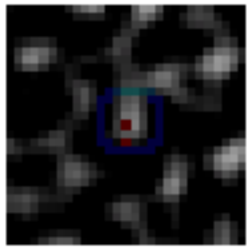
- Apply the classifier to stamps centered on any **pixel of interest** of the original SNR maps (e.g: snr H2 >= 2)

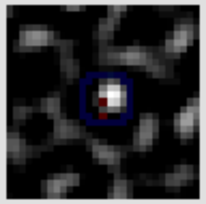
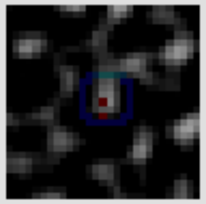
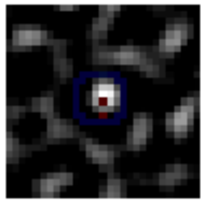
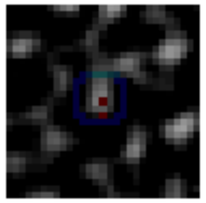
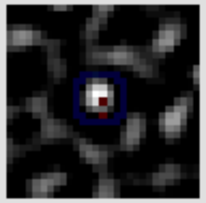
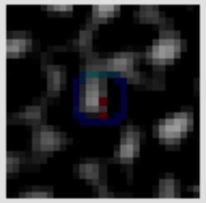
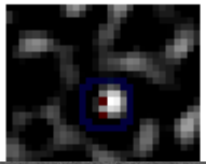
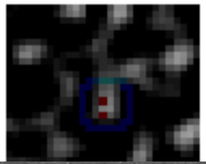
  ➡ List of candidate stamps

- Cluster the candidates by locations in the image

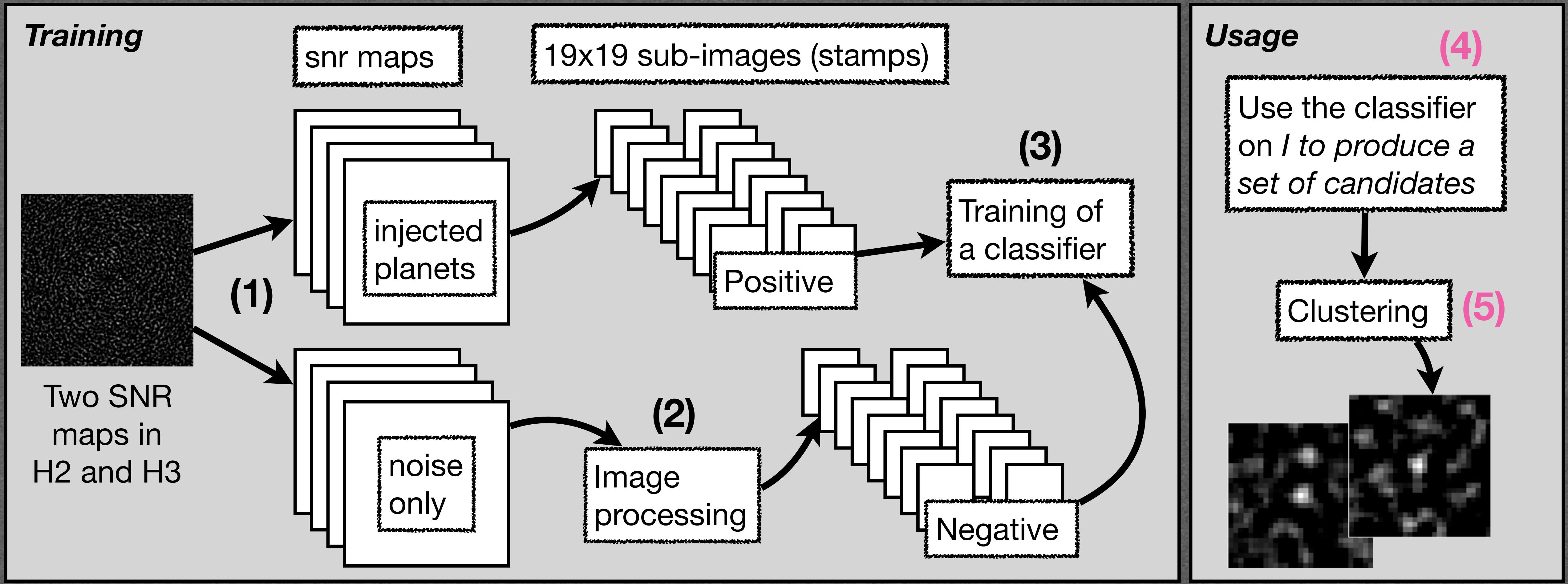
  ➡ Nearby candidates are gathered in the same cluster

(4-5)

Star Name	Name	Center SNR Max	Coordinate	Dist. Star	Prediction	Image L1	Image L2	Details
Cluster 1 - 8 stamps (No inj.) details								
51Eri	Stamp1465_D-1_L00_713_687	3.66	(713, 687)	39.0	0.83			details

Star Name	Name	Center SNR Max	Coordinate	Dist. Star	Prediction	Image L1	Image L2	Details
51Eri	Stamp1465_D-1_L00_713_687	3.66	(713, 687)	39.0	0.83			details
51Eri	Stamp1466_D-1_L00_714_687	3.66	(714, 687)	38.0	0.96			details
51Eri	Stamp1467_D-1_L00_715_687	3.66	(715, 687)	38.0	0.96			details
51Eri	Stamp1471_D-1_L00_713_688	3.66	(713, 688)	38.0	0.77			details

⋮



Current results

Current results - data sets

- Four stars:
  1. HD 108767B
  2. HIP 1993
  3. HIP 12394
  4. HIP 107345

- Two « Blind Tests » on the same 4 stars:
  1. BT1: using the snr maps computed for the SHINE blind test (injected signals **different:** SNR versus physical)
  2. BT2: designed by Antoine

- One real case study
  1. 51Eri

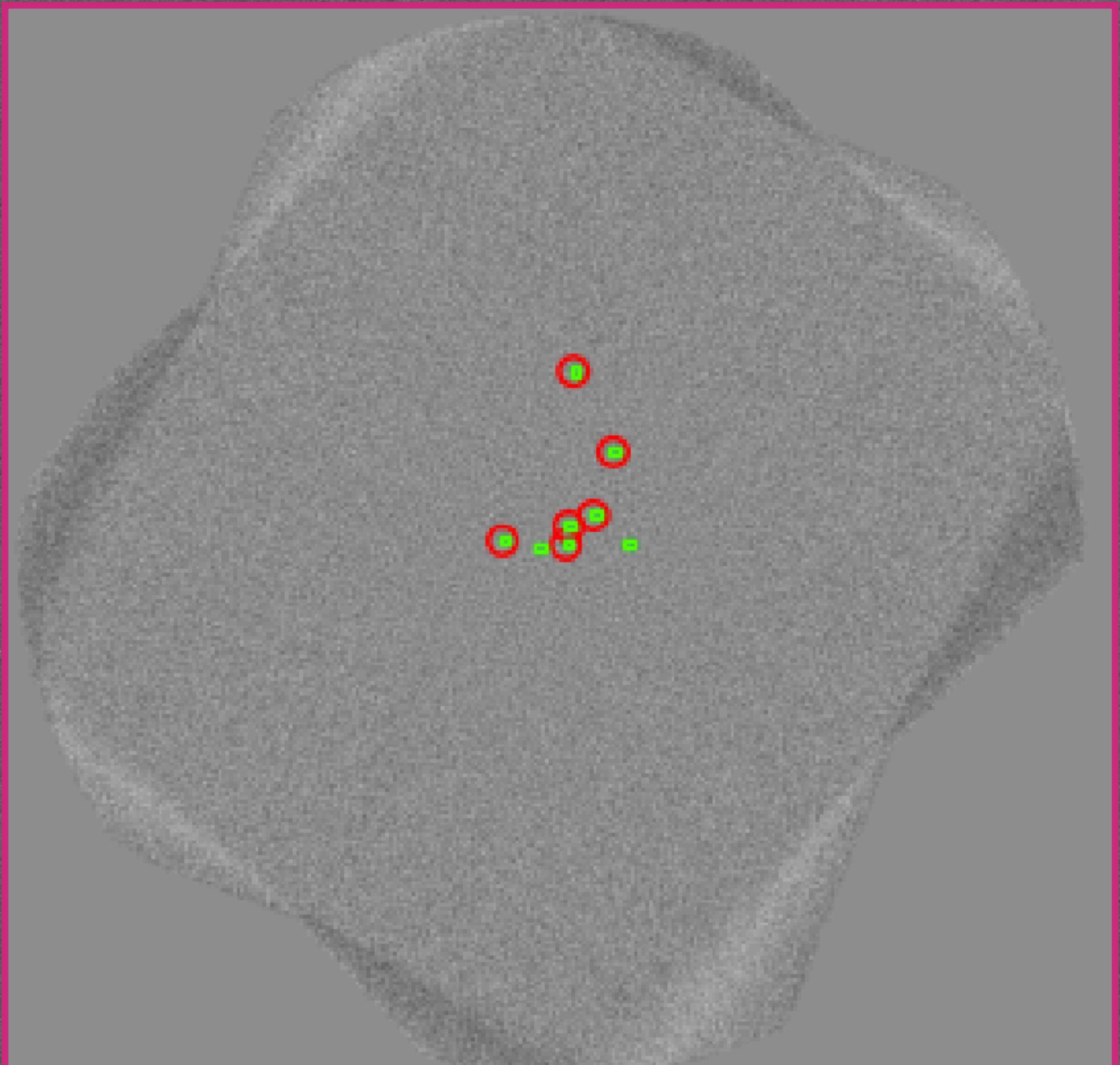
Todo: + weather conditions

star name	learning		BT1	BT2
	#pos	#neg	#pos	#pos
all	1026	77565	29	140
HD108767B	259	15621	7	26
HIP1993	253	15228	8	37
HIP12394	251	15636	6	39
HIP107345	263	31080	8	38

Current results - final performance

	BT2		
		Logistic regression	
	Nb Inj	#Found	#Cand
HD108767B	26	**22**	10
HIP1993	37	**33**	0
HIP12394	39	26	2
HIP107345	38	38	0
Total	140	119	12
	BT1		
HD108767B	7	5	11
HIP1993	8	6	0
HIP12394	6	4	2
HIP107345	8	7	1
Total	29	22	14

Table 2. Comparing the logistic regression
tection algorithm of PACO.



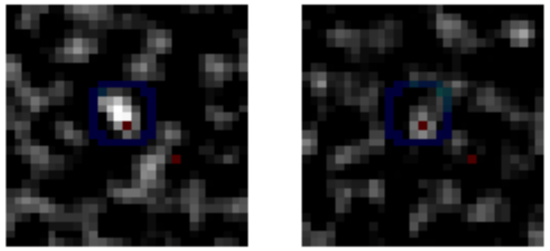
Current results - final performance

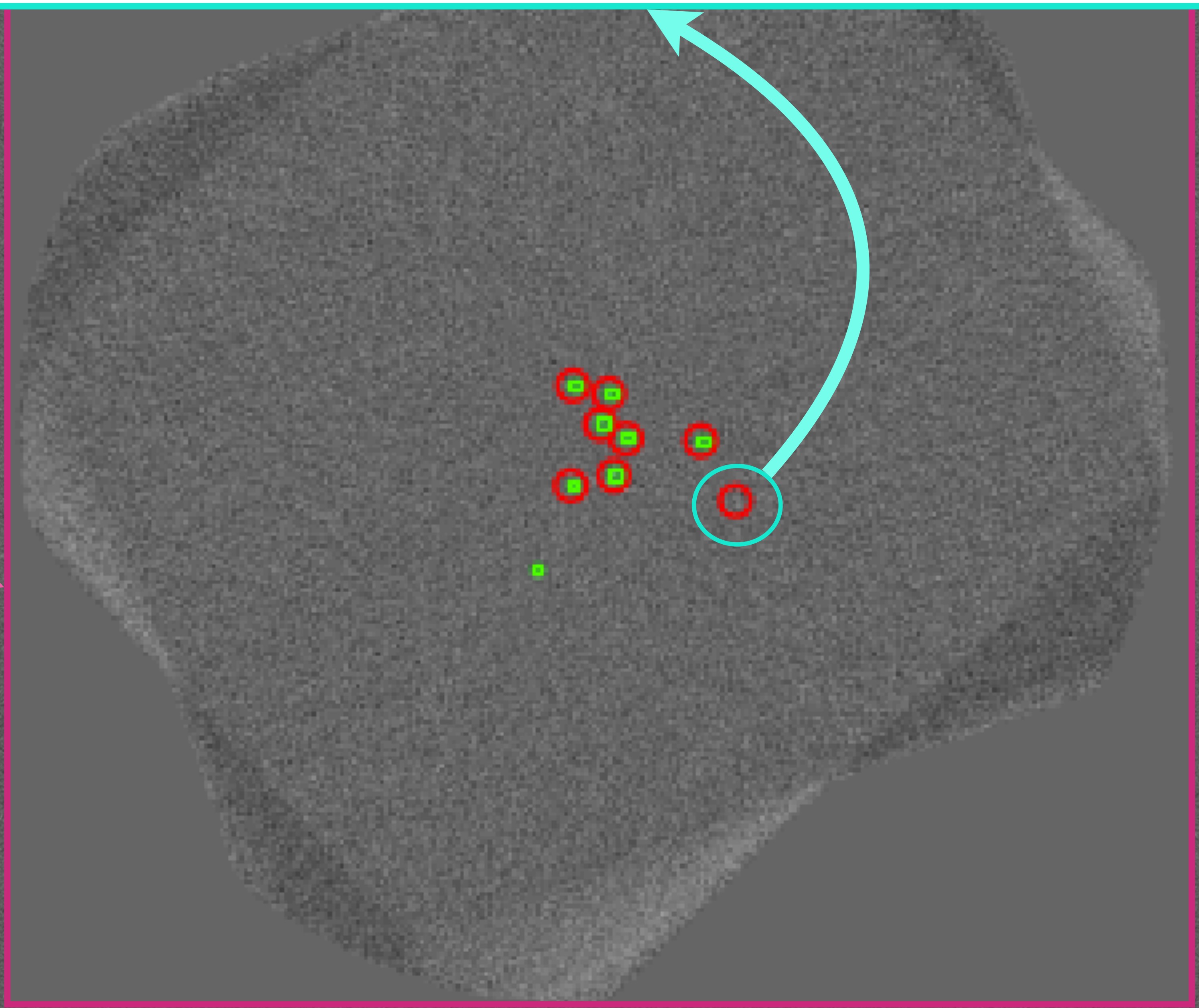
	BT2		
		Logistic regression	
	BT1		
HD108767B	7	5	11
HIP1993	8	6	0
HIP12394	6	4	2
HIP107345	8	7	1
Total	29	22	14

Table 2. Comparing the logistic regression
tection algorithm of PACO.

Cluster 7 - 8 stamps (No inj.) details

BT_HIP107345	Stamp56766_D69_L00_853_639	3.45	(853, 639)	154.0	0.7

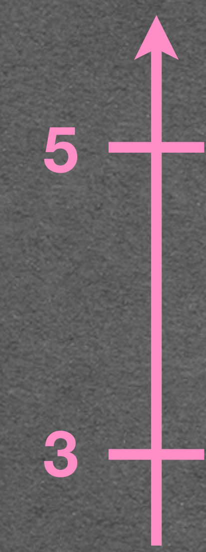




Current results - final performance

		BT2					
		Logistic regression		PACO threshold 3		PACO threshold 5	
	Nb Inj	#Found	#Cand	#Found	#Cand	#Found	#Cand
HD108767B	26	**22**	10	21	2	16	0
HIP1993	37	**33**	0	32	67	27	0
HIP12394	39	26	2	**28**	304	20	0
HIP107345	38	38	0	38	59	33	0
Total	140	119	12	119	432	96	0
		BT1					
HD108767B	7	5	11	**6**	145	3	0
HIP1993	8	6	0	**8**	85	3	0
HIP12394	6	4	2	**5**	293	2	0
HIP107345	8	7	1	**8**	61	2	0
Total	29	22	14	**27**	584	10	0

Table 2. Comparing the logistic regression approach versus the default threshold detection algorithm of PACO.



1. The classifier is « conservative » with little false positive
2. Compared to an SNR threshold of 3, the classifier misses few planets but has less false positives
3. Compared to an SNR threshold of 5, the classifier detects significantly more planets

Note on methodology: results sent to Antoine for BT2 without us knowing the locations of the planets :)

Current results - Case study 51Eri = HIP21547

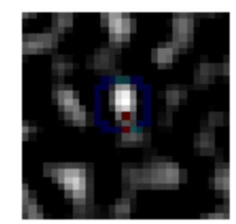
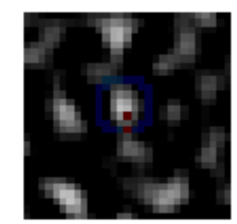
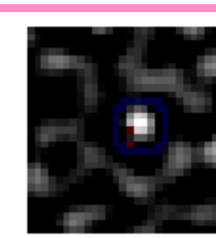
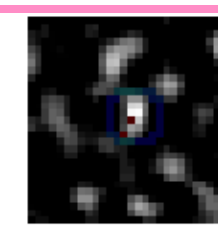
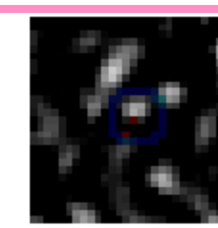
- One real case study, 4 epochs

		#Cand
1.	HIP21547_20151225	2
2.	HIP21547_20161212	3
3.	HIP21547_20160115	4
4.	HIP21547_20161211	3

- The classifier finds the planet 3 times over 4 with SNR of
  - 4.71 (20151225)
  - 5.28 (20160115)
  - 2.69 (20161212)

- 2/3 cases are below a SNR threshold of 5

Cluster 2 - 10 stamps (No inj.) details

HIP21547_20151225	Stamp9177_D-1_L00_715_686	4.71	(715, 686)	39.0	0.89		
HIP21547_20160115	Stamp17090_D-1_L00_713_687	5.28	(713, 687)	39.0	0.96		
HIP21547_20161212	Stamp4303_D-1_L00_711_688	2.69	(711, 688)	38.0	0.79		

Current results - further analysis

1. What is the **accuracy of features** i.e is the numerical value (significantly) different for negative and positive stamps ?

2. What are the features really needed to achieve the same level of performance ?

3. Performance of a **single classifier** (learnt over all snr maps) versus **many classifiers** (one for each snr map)

4. What is **the amount of data** (injections + noise map) needed to achieve the same level of performance ?

5. Tuning the prediction threshold (Precision / Recall)

Conclusion - Future work

1. A simple filtering of candidates identified by a statistical approach that avoids self-subtraction (PACO):
   - image processing (edge detection)
   - logistic regression
   - clustering

2. SNR threshold => Prediction threshold
   - limited set of features (SNR, Gradient SNR, Airy figure, Speckle)
   - suited for multi-spectral information

3. Old school type of machine learning:
   - features have direct physical/optical meaning
   - frugal algorithm (few minutes) to the exception of injections

A. More real case studies (usage on non injected planets)
B. Multi-spectral data sets (> 2 wavelengths)
C. Incorporate more **physical/optical knowledge** into features
D. How to quantify the **confidence level** ?
E. **class imbalance** and **features not defined everywhere** (e.g: speckle)
F. Machine learning where the **class is only known with a probability** ?


Nicolas
Questions

Additional results
if needed

Current results - features accuracy

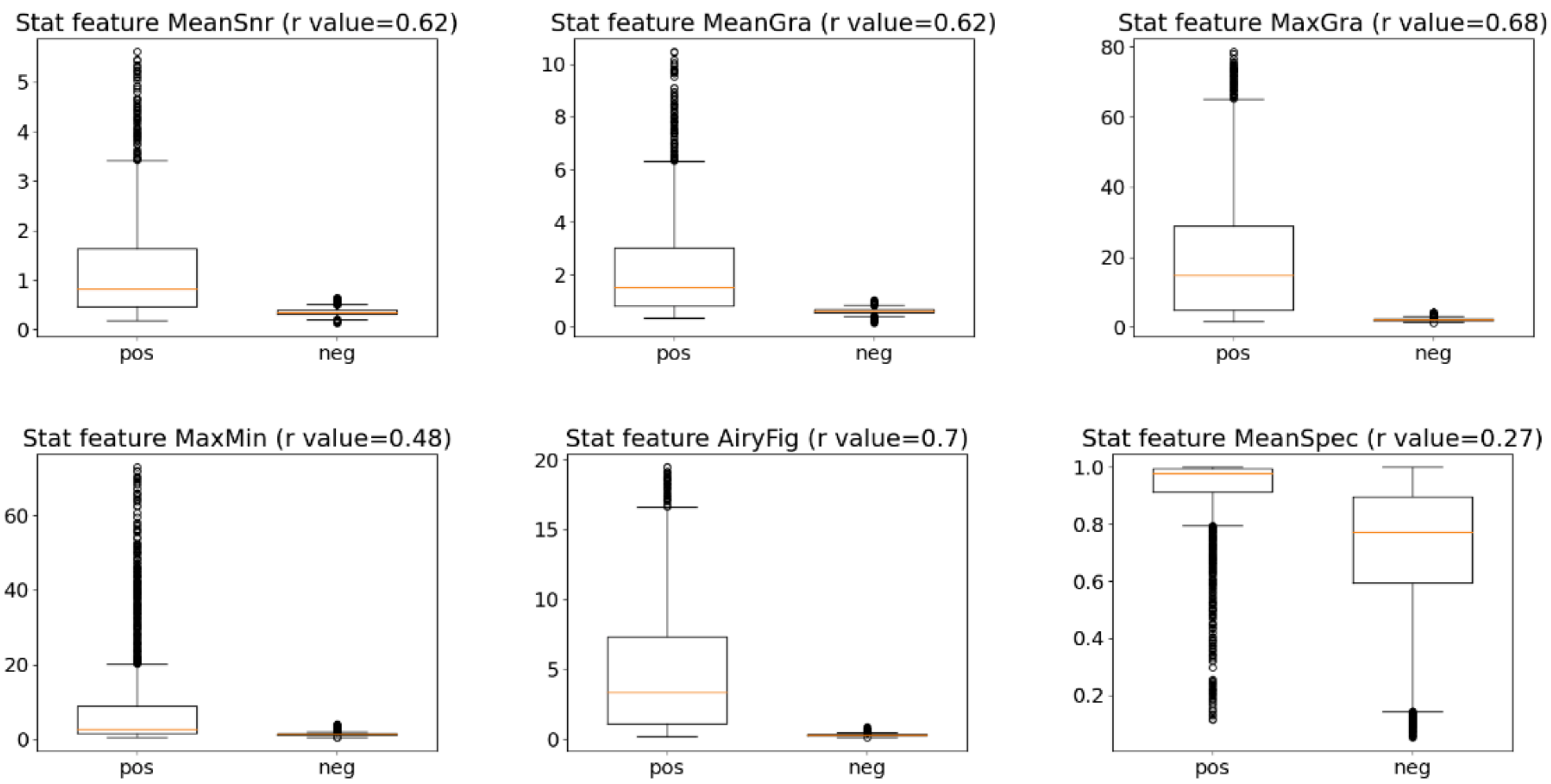


Fig. 7. Distribution of the values of 6 features computed over stamps of SNR ≥ 2 of the four main images (2069 positives, 18074 negatives).

1. Airy figure, SNR intensity, SNR gradient are similarly correlated to the class
2. The speckle feature does not seem very discriminative

Current results - features accuracy - Speckles

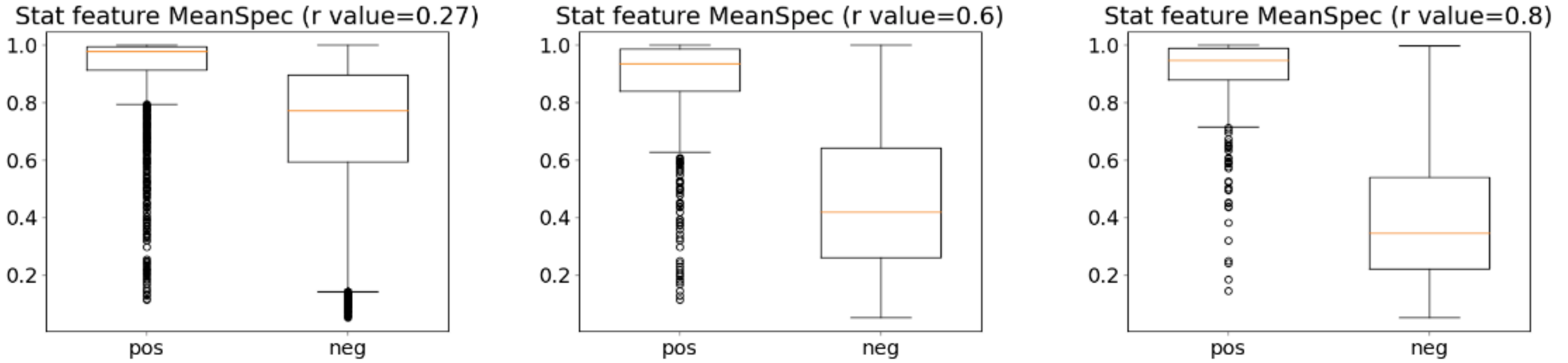


Fig. 8. Distribution of the **MeanSpec** feature for three selected subsets of stamps respectively from left to right: (1) all stamps of SNR $\geq$ 2 (2069 positives, 18074 negatives), (2) all stamps of SNR $\geq$ 2 located at a radius in [30, 140] (554 positives, 1969 negatives), (3) all stamps of SNR $\geq$ 2.5 located at a radius in [30, 140] (492 positives, 558 negatives).

1. Relevance of speckle seems to increase as the radius is narrowed to the « proper » ring around the star

Current results - single versus image dedicated classifier

1. **Image dedicated classifier**: a classifier is learnt for a given snr map and is meant to be used only on this map
2. **Single classifier**: a single classifier is learnt from all the available snr maps once and for all.

	BT2			BT1		
	#inj	#found	#cand	#inj	#found	#cand
Single classifier						
4 stars	140	**120**	18	29	**23**	23
One classifier per image						
HD108767B	26	22	10	7	5	11
HIP1993	37	33	0	8	6	0
HIP12394	39	26	2	6	4	2
HIP107345	38	38	0	8	7	1
Total	140	119	**12**	29	22	**14**

Table 3. Comparing a single classifier trained over the 4 images to a classifier dedicated to each image.

Current results - precision and recall

Precision: How many retrieved items are relevant ?

Recall: How many relevant items are retrieved ?

The « best » threshold is determined from the f-score:

$$F_\beta = (1 + \beta^2)\frac{precision \times recall}{(\beta^2 \times precision) + recall}$$

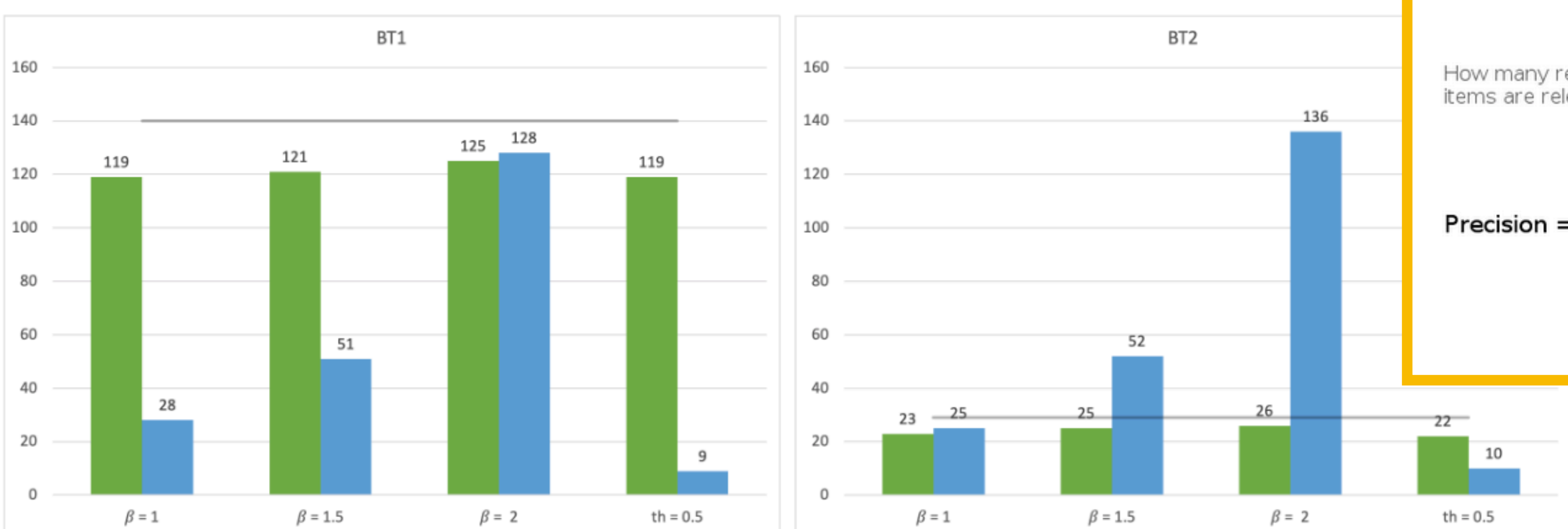
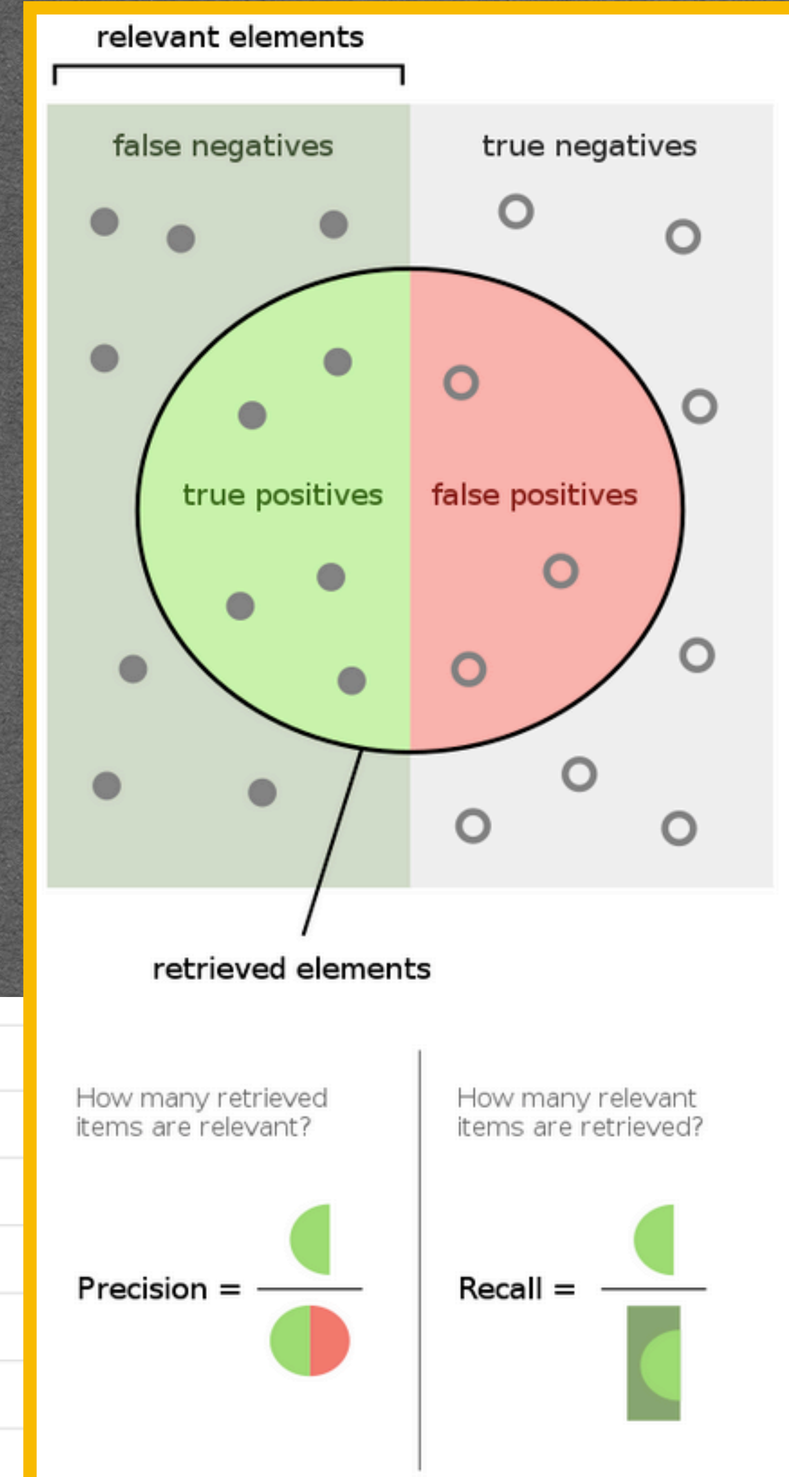


Fig. 9. Number of objects found (in green) and candidates (in blue) according to the threshold. The grey line represents the number of injections.



wikipedia