ORIGIN: Blind detection of faint emission line galaxies in MUSE datacubes

David Mary, Roland Bacon, Simon Conseil, Laure Piqueras, Antony Schutz

A&A 635, A194 (2020)
https://doi.org/10.1051/0004-6361/201937001

Astronomy & Astrophysics

ORIGIN: Blind detection of faint emission line galaxies in MUSE datacubes⋆

David Mary[1], Roland Bacon[2], Simon Conseil[2], Laure Piqueras[2], and Antony Schutz[1,2]

[1] Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire Lagrange, Bd de l'Observatoire, CS 34229, 06304 Nice Cedex 4, France
e-mail: david.mary@unice.fr
[2] Univ Lyon, Univ Lyon1, ENS de Lyon, CNRS, Centre de Recherche Astrophysique de Lyon UMR5574, 69230 Saint-Genis-Laval, France

Received 28 October 2019 / Accepted 16 January 2020

ABSTRACT

Context. One of the major science cases of the Multi Unit Spectroscopic Explorer (MUSE) integral field spectrograph is the detection of Lyman-alpha emitters at high redshifts. The on-going and planned deep fields observations will allow for one large sample of these sources. An efficient tool to perform blind detection of faint emitters in MUSE datacubes is a prerequisite of such an endeavor.
Aims. Several line detection algorithms exist but their performance during the deepest MUSE exposures is hard to quantify, in particular with respect to their actual false detection rate, or purity. The aim of this work is to design and validate an algorithm that efficiently detects faint spatial-spectral emission signatures, while allowing for a stable false detection rate over the data cube and providing in the same time an automated and reliable estimation of the purity.
Methods. The algorithm implements (i) a nuisance removal part based on a continuum subtraction combining a discrete cosine transform and an iterative principal component analysis, (ii) a detection part based on the local maxima of generalized likelihood ratio test statistics obtained for a set of spatial-spectral profiles of emission line emitters and (iii) a purity estimation part, where the proportion of true emission lines is estimated from the data itself: the distribution of the local maxima in the "noise only" configuration is estimated from that of the local minima.
Results. Results on simulated data cubes providing ground truth show that the method reaches its aims in terms of purity and completeness. When applied to the deep 30 h exposure MUSE datacube in the *Hubble* Ultra Deep Field, the algorithms allows for the confirmed detection of 133 intermediate redshifts galaxies and 248 Lyα emitters, including 86 sources with no *Hubble* Space Telescope counterpart.
Conclusions. The algorithm fulfills its aims in terms of detection power and reliability. It is consequently implemented as a Python package whose code and documentation are available on GitHub and readthedocs.

Key words. methods: data analysis – techniques: imaging spectroscopy – galaxies: high-redshift – methods: statistical

1. Introduction

Spectroscopic observations of galaxies at high redshift have recently been revolutionized by the Multi Unit Spectroscopic Explorer (MUSE) instrument in operation at the VLT (Very Large Telescope) since 2014. MUSE is an adaptive optics assisted integral field spectrograph operating in the visible (Bacon et al. 2010, 2014). With its field of view of 1×1 arcmin^2 sampled at 0.2 arcsec and its simultaneous spectral range of 4800–9300 Å at $R \sim 3000$, MUSE produces large hyperspectral datacubes of 383 millions voxels, corresponding to about $320 \times 320 \times 3680$ pixels along the spatial (x, y) and spectral (z) axes. Its unique capabilities of providing three-dimensional (3D) deep field observations have been demonstrated in the early observations of the *Hubble* Deep Field-South (Bacon et al. 2015) and more recently in the *Hubble* Ultra Deep Field (HUDF, Bacon et al. 2017) and the CANDELS – GOOD South area (Urrutia et al. 2019).

Thanks to its unrivalled capabilities, MUSE has been able to increase the number of spectroscopic redshifts in these fields by an order of magnitude (see for example Inami 2017). The most spectacular increase is at high redshift ($z > 3$), where MUSE was able to detect a large number of Lyα emitters. In the deepest exposures (10+ h), MUSE is even able to go beyond the limiting magnitude of the deepest *Hubble* Space Telescope (HST) exposures. For example in the HUDF, which achieves a 5σ depth of 29.5 in the *F775W* filter, MUSE was able to detect Lyα emitters without an HST counterpart (Bacon et al. 2017; Maseda et al. 2018). These observations have led to a breakthrough in our understanding of the high redshift Universe, which includes, for example, the discovery of Lyα emission from the circumgalactic medium around individual galaxies (Wisotzki et al. 2016, 2018; Leclercq 2017) or the role of the low mass galaxies and the evolution of the faint-end Lyα luminosity function (Drake 2017).

Building a large sample of low luminosity Lyα emitters at high redshift is an important objective for the near future with the upcoming deep fields observations currently executed or planned in the context of the MUSE guaranteed time observations (GTO). The availability of an efficient tool to perform blind detection of faint emitters in MUSE datacubes is a prerequisite of such an endeavor.

⋆ ORIGIN: https://github.com/musevlt/origin

Several tools have already been developed to perform blind searches of faint emitters in MUSE datacubes, such as MUSELET, a SExtractor based method available in MPDAF (Piqueras et al. 2017), LSDCAT, a matched filtering method (Herenz & Wisotzki 2017) or SELFI, a Bayesian method (Meillier et al. 2016). These tools have been successfully used so far, for instance LSDCAT and MUSELET in the context of respectively the MUSE Wide Survey (Urrutia et al. 2019) and the analysis of the lensing clusters (e.g., Lagattuta et al. 2019). However, none of these methods currently allow for a reliable estimate of the proportion of false discoveries (or purity) actually present in the list of detected sources. As a consequence their actual performance on the deepest MUSE exposures, for which no ground truth is available indeed, is consequently hard to quantify.

Furthermore, from our experience in the blind search of emitters in the MUSE deep fields, we have learned that when tuned to be efficient for the faintest emitters, every detection method becomes overly sensitive to any residuals left by the imperfect continuum subtraction of bright sources and by the data reduction pipeline (e.g., instrumental signatures or sky subtraction residuals). This leads to a global inflation of the false detections, at levels that are unpredictable and fluctuating in the datacube. This effect requires either to limit the detection power by setting a threshold high enough to stay on the "safe side", or to consent spending a significant human-expert time to examine the long list of potential discoveries proposed by the algorithm.

In this context, we have developed an automated method, called ORIGIN, allowing for these methodological and computational challenges. In this paper, we present in detail the algorithm. An overview is given in Sect. 2 and a step-by-step description in Sect. 3. The application of ORIGIN to the deep 30 h exposure MUSE datacube in the HUDF called udf-10 is presented in Sect. 4. Mathematical complements, implementation of the method into a publicly released software and parameters values used for the data cube udf-10 are given the Appendices. Conclusions and possible improvements are listed in the last section.

2. Overview

2.1. Notations

In the following, we note N_x, N_y and N_λ the two spatial and the spectral dimensions of the data cube. (For udf-10 this corresponds to $322 \times 323 \times 3681 \approx 383$ millions voxels.) Bold lowercase letters as $\mathbf{x}$ denote vectors and bold capital letters as $\mathbf{X}$ denote matrices. We also use the following notations:

– $\mathbf{s}_i = [s_1, \ldots, s_{N_\lambda}]^\top$ is one spectrum in the data cube, with $i \in \{1, \ldots, N_x \times N_y\}$.

– A collection of spectra $\{\mathbf{s}_i\}$ from a data cube can be stored in a matrix whose columns are the spectra. For the whole data cube, this matrix is noted $\mathbf{S} := [\mathbf{s}_1, \ldots, \mathbf{s}_{N_x \times N_y}]$.

– $\mathbf{\Sigma}_i$ is the covariance matrix of spectrum $\mathbf{s}_i$. It is provided by the data reduction pipeline.

– Whitened (or standardized) vectors and matrices are denoted with a $\widetilde{\ }$. For instance : $\widetilde{\mathbf{s}_i} := \mathbf{\Sigma}_i^{-\frac{1}{2}} \mathbf{s}_i$.

– $\mathbf{0}_{N_x,N_y,N_z}$ and $\mathbf{1}_{N_x,N_y}$ represent respectively arrays of zeroes and ones of size $N_x \times N_y \times N_z$ and $N_x \times N_y$.

– The symbols $\otimes$ and $\oslash$ represent respectively convolution and pointwise division.

– For a cube stored in a matrix $\mathbf{A}$, notations $\mathbf{A}(\cdot, \cdot, k)$ and $\mathbf{A}(i, j, \cdot)$ represent respectively the image ("slice") of the cube in the kth wavelength band and the spectrum of the cube at spatial positions (i, j).

For simplicity, we often drop index i when the described processing is applied sequentially to all spectra.

2.2. Objectives

The detection algorithm ORIGIN is aimed at detecting point-like or weakly resolved emission line sources with emission lines at unknown redshifts. The algorithm was designed with a threefold objective : (1) Be as powerful as possible for detecting faint emission lines; (2) Be robust to local variations of the data statistics, caused for instance by bright, extended sources, residual instrumental artifacts or different number of co-added exposures; (3) Provide a list of detected sources that achieves a target purity, this target purity being specified by the user. These objectives led us to a detection approach in five main stages outlined below and described step by step in Sect. 3.

2.3. Overview

The flowchart in Fig. 1 shows an overview of the algorithm. We give here a brief summary of each step (purple boxes), a detailed description of which can be found in the Sections indicated in these boxes.

2.3.1. Nuisance removal and segmentation

Spatial and spectral components that are not emission lines in the datacube $\mathbf{S}$ (e.g., bright sources, diffuse sources, instrumental artifacts) can drastically increase the false detection rate. The first step is thus to detect and to remove such sources, called "nuisances" below. Regions of the field that are free from such sources are called "background". Nuisance features are removed by combining a continuum removal (using a discrete cosine transform, DCT) and an iterative principal component analysis (PCA). The estimated continuum ($\mathbf{C}$) and the residual data cubes ($\mathbf{R} = \mathbf{S} - \widehat{\mathbf{C}}$) from the DCT are used to perform a spatial segmentation. This provides for each spectrum a label (nuisance or background. These labels are stored in a matrix ($\mathbf{L}$) and are used by the PCA, which acts differently on background and nuisance spectra.

2.3.2. Test statistics

In the faint residual datacube ($\mathbf{F}$) the point is now to capture the signatures of line emissions. These signatures essentially take the form of "3D blobs" corresponding to the convolution of a spectral line profile by the spatial PSF of the instrument. Several matched filtering (Herenz & Wisotzki 2017) or Generalized Likelihood Ratio (GLR) approaches (Paris et al. 2013; Suleiman et al. 2013, 2014) have been devised for that purpose. Such approaches lead to filter the data cube with a library of possible signatures, built as spectral line profiles spread spatially by the PSF, and to normalize the result.

The filtering and normalization process considered here is also derived from a GLR approach but it is robust to an unknown residual continuum (see Sect. 3.2). It produces two data cubes ($\mathbf{T}^+_{\mathrm{GLR}}$ and $\mathbf{T}^-_{\mathrm{GLR}}$), which correspond respectively to intermediate tests statistics obtained when looking for emission and absorption lines. When present in the data, an emission line emitter tends to increase locally the values in $\mathbf{T}^+_{\mathrm{GLR}}$, with a local maximum close to the actual position (in x, y and z coordinates) of the line. These local maxima are used as final test statistics for line detection: each local maximum above a given threshold is

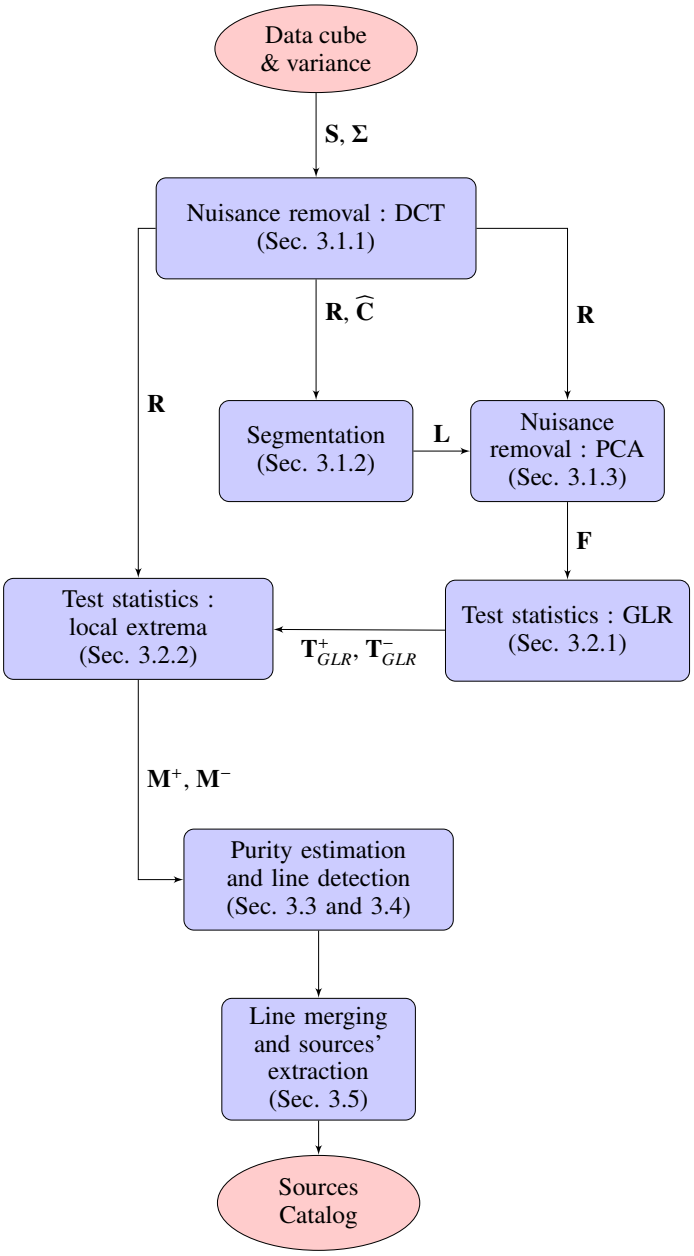

Fig. 1. Overview of the ORIGIN algorithm. A session example is given in Appendix B and a list of the main parameters in Appendix C.

identified as a line detection. It is important to note that for simplicity $\mathbf{T}^-_{\mathrm{GLR}}$ is computed so that an absorption also appears as a local maximum in $\mathbf{T}^-_{\mathrm{GLR}}$, so that the final test statistics are obtained as the local maxima (called $\mathbf{M}^+$ and $\mathbf{M}^-$) of $\mathbf{T}^+_{\mathrm{GLR}}$ and $\mathbf{T}^-_{\mathrm{GLR}}$.

2.3.3. Purity estimation

Evaluating the purity of the detection results, that is, the fraction of true sources in the total list of detected sources, requires to estimate the number of local maxima that would be larger than the threshold "by chance", that is, in absence of any emission line. The complexity of the data prevents us from looking for a single, reliable analytical expressions of this distribution. However, the size of the data cube and the nature of our targets (emission lines) allows for the estimation of this distribution from the data – an approach advocated for instance by Walter et al. (2016) and Bacher et al. (2017). When the data contains only noise and under mild assumption on this noise, the statistics obtained when looking for emission lines are the same as those obtained when looking for absorption lines: the local maxima of $\mathbf{M}^+$ and $\mathbf{M}-$ should have the same distribution. Hence, the number of local maxima of $\mathbf{M}^+$ that should be above any given threshold γ under the null hypothesis can be estimated from the number of local maxima found above this threshold in $\mathbf{M}^-$. This provides an estimate of the false discoveries that should be expected for any threshold, and hence of the purity. In practice, this estimation is done for a grid of threshold values. The value of the threshold corresponding to the purity desired by the user is identified and the emission lines correspond to the local maxima of $\mathbf{M}^+$ exceeding this threshold.

2.3.4. Detection of bright emission lines

As explained in Sect. 2.3.1 the iterative PCA aims at removing all features that do not correspond to faint emission lines. This means that bright emission lines – the most easily detectable ones – can be removed by the PCA step and further be missed by the detection process. It is thus necessary to detect such lines before the PCA, in the DCT residual $\mathbf{R}$. The procedure for detecting these lines and controlling the purity of this detection step strictly mirrors what is done for faint lines, with local maxima computed directly from (whitened versions of) $\mathbf{R}$ and $-\mathbf{R}$ instead of $\mathbf{T}^+_{\mathrm{GLR}}$ and $\mathbf{T}^-_{\mathrm{GLR}}$.

2.3.5. Line merging and sources' extraction

The detected bright and faint emission lines are finally merged into sources, for which several informations (refined position, spectral profile and total flux) are computed and stored in the final source catalog (cf. Sect. 3.5).

3. Step-by-step method description

3.1. Nuisance removal and spatial segmentation

The strategy of ORIGIN is to track and suppress nuisance sources while preserving the targets, the line emitters. The test statistics computed from the residuals still have to be robust against unknown fluctuating flux variations under the null hypothesis, but only moderately so if the nuisance removal is efficient (cf. Sect. 3.2).

The removal of nuisance sources is performed spectrum-wise in two steps explained below: DCT and iterative PCA. The first stage of DCT (a systematic and fast procedure) helps to remove quickly energetic and smooth fluctuations. In contrast, the version of the iterative PCA designed for this problem can capture adaptively signatures that are much fainter and possibly very complex in shape, a but it is computationally heavy. This combination makes the overall nuisance removal process efficient and tractable in a reasonable time.

3.1.1. Nuisance removal with DCT

Each spectrum $\mathbf{s}$ is modeled as

$$\mathbf{s} = \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{1}$$

where $\mathbf{D}$ is a partial DCT matrix of size $N_\lambda \times N_{\mathrm{DCT}}$, $\boldsymbol{\alpha}$ is a $N_{\mathrm{DCT}} \times 1$ vector of decomposition coefficients, $\mathbf{D}\boldsymbol{\alpha}$ is the unknown

continuum and $\boldsymbol{\epsilon}$ is an additive Gaussian noise with covariance matrix $\boldsymbol{\Sigma}$. Maximum Likelihood estimation of the continuum of spectrum $\mathbf{s}$ leads to a weighted least squares problem, for which the estimated coefficients $\widehat{\boldsymbol{\alpha}}$ are obtained by (cf. Sect. 2.1 for the $\widetilde{\ }$ notation):

$$\widehat{\boldsymbol{\alpha}} := \arg\min_{\boldsymbol{\alpha}} \|\widetilde{\boldsymbol{s}} - \widetilde{\boldsymbol{D}}\boldsymbol{\alpha}\|^2 = (\widetilde{\boldsymbol{D}}^\top \widetilde{\boldsymbol{D}})^{-1} \widetilde{\boldsymbol{D}}^\top \widetilde{\boldsymbol{s}}. \tag{2}$$

The estimated continuum $\hat{\boldsymbol{c}}$ and residual $\mathbf{r}$ are obtained by

$$\begin{cases} \hat{\boldsymbol{c}} = \mathbf{D}\widehat{\boldsymbol{\alpha}}, \\ \mathbf{r} = \mathbf{s} - \hat{\mathbf{c}}. \end{cases} \tag{3}$$

These spectra are collected in continuum and residual data cubes named $\widehat{\mathbf{C}}$ and $\mathbf{R}$ respectively. The parameter N_{DCT} controls the number of DCT basis vectors used in the continuum estimation. A value that is too small lets large scale oscillations in the residual spectrum, while a value that is too large tends to capture spectral features with small extension like emission lines, which become then more difficult to detect in the residual. A satisfactory compromise was found here[1] with $N_{\mathrm{DCT}} = 10$. This value leaves the lines almost intact: typically, the energy of the line in the DCT residual remains close to 100% until N_{DCT} reaches several hundreds, depending on the line width. The continuum subtraction with $N_{\mathrm{DCT}} = 10$ is not perfect, but a large part of the work is done: for bright objects, 99 % of the continuum's energy is typically contained in the subspace spanned by the first 10 DCT modes and decreases very slightly afterward. The PCA does the rest.

Before describing the PCA we present a segmentation step, which is required to implement the PCA.

3.1.2. Spatial segmentation

The purpose of spatial segmentation is to locate regions where the data spectra contain nuisance sources and where they are free from them (in which case they are labeled as "background"). This segmentation is necessary to further remove the nuisances. As mentioned before, such spectra can have residuals from continuum or bright emission lines, or correspond to regions exhibiting a particular statistical behavior, caused for instance by the presence of instrumental artifacts or residuals from sky subtraction . The segmentation step relies both on the information extracted in the continuum cube $\widehat{\mathbf{C}}$ and on the residual cube $\widetilde{\mathbf{R}}$ (which is whitened in order to account for the spectral dependence of the noise variance). In $\widehat{\mathbf{C}}$, an energetic spectrum $\widehat{\mathbf{c}}$ indicates the presence of a continuum. In $\widetilde{\mathbf{R}}$, a spectrum $\widetilde{\mathbf{r}}$ containing residual signatures from bright sources or artifacts tends to have higher energy than pure noise (background) pixels. For these reasons, we found that the following two tests statistics are both efficient and complementary to locate nuisance sources:

$$\begin{cases} t_1(\widehat{\mathbf{c}}) := \log_{10} \|\widehat{\mathbf{c}}\|^2, \\ t_2(\widetilde{\mathbf{r}}) := \frac{1}{N_\lambda} \|\widetilde{\mathbf{r}}\|^2. \end{cases} \tag{4}$$

For a spectrum under test $\mathbf{x}$, the segmentation tests are both of the form

$$t(\mathbf{x}) \underset{\text{background}}{\overset{\text{nuisance}}{\gtrless}} \gamma, \tag{5}$$

where t is either t_1 or t_2 in (4), $\mathbf{x}$ is either $\widehat{\mathbf{c}}$ or $\widetilde{\mathbf{r}}$ and γ is a threshold allowing for the tuning of the sensitivity of the tests[2]. In the data, the spectrum at spatial coordinates (i, j) in the field is considered containing nuisances if at least one of the two tests applied to the corresponding spectra $\widehat{\mathbf{c}}$ or $\widetilde{\mathbf{r}}$ leads to a result above the test threshold. The $\log_{10}$ function in the definition of t_1 in (4) is there to stabilize numerically the test statistic, as the squared norm of the estimated continuum may vary by several orders of magnitudes within a data cube.

In data like udf-10, both distributions appear to be right-skewed, with a roughly Gaussian left tail and a much heavier, possibly irregular right tail caused by nuisance sources. Hence, the distribution of the test statistics that would be obtained with pure noise (without nuisances) is estimated by means of the left part of the empirical distribution, for which a Gaussian approximation provides a reasonable fit (see Fig. 3). This leads to an estimated distribution of the test statistics under the noise only hypothesis which is Gaussian with estimated mean $\widehat{\mu}(t)$ and estimated standard deviation $\widehat{\sigma}(t)$. For the purpose of segmentation (or classification), the user chooses a level of error in the form of an error probability, called $P_{\mathrm{FA}}^{\mathrm{seg}}$ below. In order to tune the tests (5) at this target error rate, one needs to find the threshold value γ such that

$$\Pr\,(t > \gamma \mid \text{noise only}) = P_{FA}^{\mathrm{seg}},$$

The higher the value of $P_{\mathrm{FA}}^{\mathrm{seg}}$, the lower the value of γ and the larger the number of spectra wrongly classified as containing nuisances. If we denote by Φ the Cumulative Distribution Function (CDF) of a standard normal variable, the threshold γ for t_1 is given by

$$\gamma(t) = \widehat{\mu}(t) + \widehat{\sigma}(t) \cdot \Phi^{-1}(1 - P_{FA}^{seg}), \tag{6}$$

where again t is either t_1 or t_2. Two segmented maps are obtained from thresholding the maps of the $N_x \times N_y$ test statistics at the values $\gamma(t_1)$ and $\gamma(t_2)$ defined above. The nuisances regions of each map are merged into a single merged segmentation map.

Because MUSE data cubes are large, a PCA working on the full data cube would be sub-optimal. Indeed, the repeated computation of the eigenvectors of a matrix composed by the whole cube would be computationally prohibitive. Moreover, the aim of the PCA is to capture spectral features corresponding to nuisances. Such nuisances have features that are locally similar, so a PCA working on patches smaller than the whole cube is also more efficient to remove the nuisances. When building these patches (whose size is typically one tenth of the whole data cube for udf-10), care must be taken that regions identified as nuisances in the previous step are not split over two such patches. For udf-10 the segmentation algorithm starts with $N_Z = 9$ rectangular patches. The nuisance zones intersecting several such patches are iteratively identified and attached to one patch, under the constraint that the final patches have surface in a target range. The minimal and maximal surfaces allowed for patches of udf-10 are respectively $S_{\min} = 80^2$ and $S_{\max} = 120^2$ pixels^2. The results of the segmentation nuisance versus background for udf-10 after merging the two maps based on t_1 and t_2 is shown in Fig. 2. The figure also shows the segmentation result into a number of $N_Z = 9$ large patches.

[1] Note that the same trade-off must be accounted for when choosing the window length for a median filter, for instance.

[2] For subsequent use (Algorithm 1), we note $\mathbf{t_1}(\mathbf{X})$ the vector collecting the test statistics of test t_1 applied to all spectra $\mathbf{x}$ of cube $\mathbf{X}$ (and similarly for $\mathbf{t_2}(\mathbf{X})$).

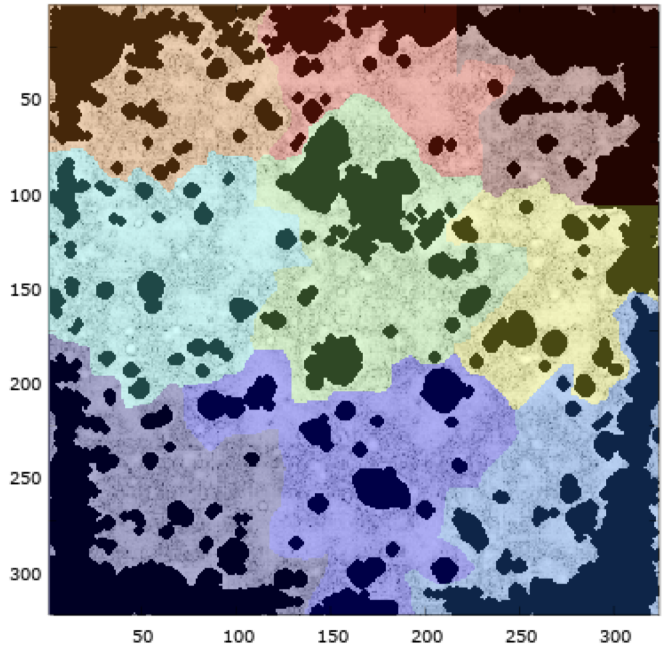

Fig. 2. Signal to noise ratio (S/N) image (sum over the wavelength channels of the raw data cube divided by the standard deviation of the noise in each voxel) with, overlaid in black, the zones classified as nuisances by test t_1 and t_2. The large patches considered for the PCA removal are shown in color.

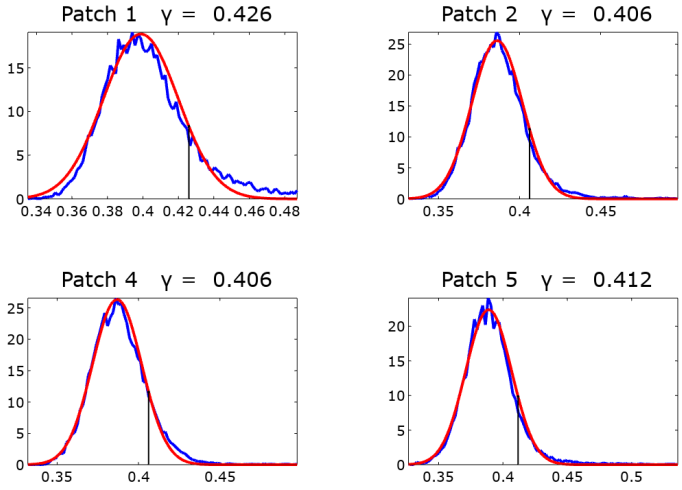

Fig. 3. Distribution of the test statistics t_2 for four patches of $\widetilde{\mathbf{R}}$ (blue), Gaussian approximation (red) and thresholds $\gamma(t_2)$ (in the titles and black stems) above which all spectra of the patch are classified as nuisance for $P_{\rm FA}^{\rm PCA} = 0.1$. The plots for the five other patches are very similar and not shown.

3.1.3. Nuisance removal with iterative PCA

The goal of this step is to iteratively locate and remove from $\widetilde{\mathbf{R}}$ residual nuisance sources, that is, any signal that is not the signature of a faint, spatially unresolved emission line. The algorithm cleans the cube one patch at a time. In order to leave in the cleaned data cube a structure compatible with noise, the nuisances are constrained to leave outside a "noise subspace", which is defined as the sample average of a small fraction (F_b) of the spectra flagged as "background" in the patch.

The Algorithm works as follows (see the pseudo-code in Algorithm 1). At each iteration, the algorithm classifies spectra of the patch under consideration (Z_i) as background or nuisance and stores them in matrices respectively called $\mathbf{B}$ and $\mathbf{N}$. This classification is done by means of the test t_2 in (4)–(5), applied to all the spectra in the patch Z_i. For this test a value $P_{\rm FA}^{\rm PCA}$ is chosen by the user, and the corresponding test threshold is computed as in (6) with $P_{\rm FA}^{\rm PCA}$ replacing $P_{\rm FA}^{\rm seg}$. In practice, for fields relatively empty as udf-10, values of $P_{\rm FA}^{\rm PCA}$ and $P_{\rm FA}^{\rm seg}$ in the range $[0.1, 0.2]$ typically provide a good trade-off in cleaning nuisances without impacting Lyα emission lines. Figure 3 shows in black the value of the threshold for four patches Z_i of the udf-10 data cube.

From the fraction (F_b %) of the spectra that show the lowest test statistics, a mean background $\overline{\mathbf{b}}$ is estimated and all the nuisance spectra in $\mathbf{N}$ are orthogonalized with respect to this background. This results in a matrix of spectra $\overline{\mathbf{N}}$, of which the first eigenvector is computed. The contribution of this vector to all spectra of the patch is removed. If some of the resulting residual spectra are still classified as nuisances, the process is repeated until there is no more spectrum classified as nuisance in the patch.

Figure 4 shows how many times each spectrum was classified as nuisance during the PCA cleaning stage. Comparing with the S/N image (gray and black zones in Fig. 2), this iteration map clearly evidences regions where bright sources and artifacts are present (e.g., horizontal lines close to the borders in the top and bottom right corner). Note also that the process is relatively fast with less than 40 iterations required in each patch to converge to a cleaned data cube.

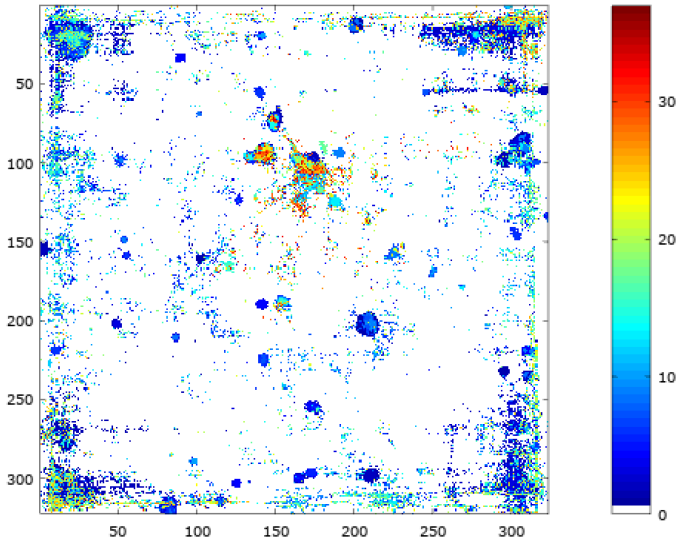

Fig. 4. Iteration map of greedy PCA for udf-10 with $P_{\rm FA}^{\rm PCA} = 0.1$. The map shows how many times each spectrum was selected in the nuisance matrix $\mathbf{N}$ before all spectra were considered "cleaned" (that is, were classified as background by test t_2).

3.2. Test statistics

3.2.1. Generalized Likelihood Ratio

For all positions (x, y, z) in the PCA residual data cube $\mathbf{F}$ (for "faint"), the algorithm considers sub-cubes $\mathbf{f}(x, y, z)$ of $\mathbf{F}$ (called $\mathbf{f}$ for short below), centered on position (x, y, z) and having the size of the considered target signatures. For each such subcube we formulate a binary hypothesis test between

– the null hypothesis: there is no emission line centred at position (x, y, z),

Algorithm 1: Iterative greedy PCA algorithm

Inputs : $\widetilde{\mathbf{R}}$, F_b, P_{FA}^{PCA}, empty data cube $\mathbf{F}$.

Output: "Cleaned" data cube $\mathbf{F}$.

for $i \leftarrow 1$ **to** N_z **do**
  $\mathbf{F}_i \leftarrow$ set of spectra of $\mathbf{F}$ in zone i
  $\mathbf{X} \leftarrow$ set of spectra of $\widetilde{R}$ in zone i
  Compute $\mathbf{B}$ and $\mathbf{N}$ using $\mathbf{t}_2(\mathbf{X})$ as in (4)
  while $\mathbf{N}$ *is not empty* **do**
    Estimate $\overline{\mathbf{b}}$ as the mean of the F_b % of the spectra having the lowest test statistics in $\mathbf{t}_2$
    $\perp \mathbf{N}$ with respect to $\overline{\mathbf{b}} : \overline{\mathbf{N}} \leftarrow \mathbf{N} - \frac{\overline{\mathbf{b}}\,\overline{\mathbf{b}}^\top}{\overline{\mathbf{b}}^\top\overline{\mathbf{b}}}\mathbf{N}$
    Compute the first eigenvector $\mathbf{v}$ of $\overline{\mathbf{N}}$
    $\perp \mathbf{X}$ with respect to $\mathbf{v} : \mathbf{X} \leftarrow \mathbf{X} - \mathbf{v}\,\mathbf{v}^\top\mathbf{X}$
    Compute $\mathbf{B}$ and $\mathbf{N}$ using $\mathbf{t}_2(\mathbf{X})$ as in (4)
  end
  Spectra of $\mathbf{F}_i \leftarrow \mathbf{X}$
end

– the alternative hypothesis: there is one emission line $\mathbf{d}_i$ centred at position (x, y, z), where $\mathbf{d}_i$ is a "3D" (spatial-spectral) signature among a set of N_s possible line signatures.

The statistical model retained to describe these two hypotheses is important, as it should capture as reliably as possible the statistical distribution of the data. This distribution results from a long chain of preprocessing, from the data reduction pipeline to the PCA described in the previous step, and may thus be very complex. On the other hand, a too sophisticated model may lead to untractable processing because it requires, in a GLR approach, maximum likelihood estimation of the corresponding unknown parameters for 380 millions positions in the datacube. We compared the performances of several statistical models and opted for the following, which allows for a good compromise between computational efficiency, detection power and robustness to remaining faint artifacts:

$$\begin{cases} \mathcal{H}_0 : \mathbf{f} = a\mathbf{1} + \mathbf{n}, \\ \mathcal{H}_1 : \mathbf{f} = a\mathbf{1} + b\mathbf{d}_i + \mathbf{n}, \ \text{with } i \in \{1, \ldots, N_s\} \text{ unknown}, \end{cases} \tag{7}$$

where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the noise assumed to be zero mean Gaussian with Identity covariance matrix under both hypotheses. The term $a\mathbf{1}$, with $a \in \mathbb{R}$ and $\mathbf{1}$ a vectorized cube of ones, models a possible residual and unknown nuisance flux, that is considered spatially and spectrally constant in subcube $\mathbf{f}$. The term $b\mathbf{d}_i$, with $b > 0$ and $i \in \{1, \ldots, N_s\}$, corresponds to one of the possible emission signatures. Each signature is a spectral profile of a given width, spatially spread in each channel by the PSF in this channel. The considered signatures define the size of $\mathbf{f}$. Spatially, they have the size of the PSF (25×25 for udf-10). Spectrally, N_s sizes are considered. For the presented udf-10 analysis we used $N_s = 3$ Gaussian profiles with FWHM of 2, 6.7 and 12 spectral channels, covering respectively 5, 20 and 31 spectral channels in total.

For binary hypothesis testing involving unknown parameters, the GLR approach (Scharf & Friedlander 1994; Kay 1998) forms the test statistics by plugging in the likelihood ratio the maximum likelihood estimates of these parameters (namely, a under $\mathcal{H}_0$ and $\{a, b, i\}$ under $\mathcal{H}_1$). This leads for each subcube $\mathbf{f}$ to a test statistics in the form of a matched filter (see Appendix A.1):

$$T^+_{\mathrm{GLR}}(\mathbf{f}) := \max_i \left(0, \frac{\mathbf{f}^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|}\right), \tag{8}$$

Algorithm 2: Computation of the test statistics

Inputs : $\mathbf{F}$, N_x, N_y, N_z, N_s, dictionary of PSF $\{\mathbf{p}_i\}$, with $i = 1, \ldots, N_z$, dictionary of spectral profiles $\{\mathbf{d}\}_i$ with $i = 1, \ldots, N_p$.

Output: $\mathbf{T}^+_{\mathrm{GLR}}$, $\mathbf{T}^-_{\mathrm{GLR}}$, $\mathbf{M}^+$ and $\mathbf{M}^-$.

1 Define $\mathbf{T}^+_{\mathrm{GLR}} = \mathbf{T}^-_{\mathrm{GLR}} = \mathbf{0}_{N_x,N_y,N_z}$
2 **for** $i \leftarrow 1$ **to** N_z **do**
3 Subtract the mean m_i of $\mathbf{p}_i$ to $\mathbf{p}_i$: $\overline{\mathbf{p}}_i \leftarrow \mathbf{p}_i - m_i\mathbf{1}$
4 Compute: $\|\overline{\mathbf{p}}_i\|^2 \leftarrow \overline{\mathbf{p}}_i^\top \overline{\mathbf{p}}_i$
5 $\mathbf{W}(\cdot, \cdot, i) \leftarrow \|\overline{\mathbf{p}}_i\|^2 \mathbf{1}_{N_x,N_y}$
6 Convolve band i of $\mathbf{F}$ with $\overline{\mathbf{p}}_i$ and store in $\mathbf{T}$: $\mathbf{T}(\cdot, \cdot, i) = \mathbf{F}(\cdot, \cdot, i) \otimes \overline{\mathbf{p}}_i$
7 **end**
8 **for** $i \leftarrow 1$ **to** N_s **do**
9 Subtract the mean m_i of $\mathbf{d}_i$ to $\mathbf{d}_i$: $\overline{\mathbf{d}}_i \leftarrow \mathbf{d}_i - m_i\mathbf{1}$
10 Compute the squared zero-mean profile, $\mathbf{d}_i^s(n)$: $\mathbf{d}_i^s(n) \leftarrow \mathbf{d}_i^2(n)$ for all entries $\mathbf{d}_i(n)$ of $\mathbf{d}_i$
11 **for** $j \leftarrow 1$ **to** N_x **do**
12 **for** $k \leftarrow 1$ **to** N_y **do**
13 $\mathbf{T}(i, j, \cdot) \leftarrow \mathbf{T}(i, j, \cdot) \otimes \overline{\mathbf{d}}_i$
14 $\mathbf{W}(i, j, \cdot) \leftarrow \mathbf{W}(i, j, \cdot) \otimes \overline{\mathbf{d}}_i^s$
15 **end**
16 **end**
17 Normalize: $\mathbf{T} \leftarrow \mathbf{T} \oslash \mathbf{W}$
18 **for** all voxels (i, j, k) **do**
19 **if** $\mathbf{T} > \mathbf{T}^+_{\mathrm{GLR}}(i, j, k)$ **then**
20 $\mathbf{T}^+_{\mathrm{GLR}}(i, j, k) \leftarrow \mathbf{T}(i, j, k)$
21 **end**
22 **if** $\mathbf{T} < \mathbf{T}^-_{\mathrm{GLR}}(i, j, k)$ **then**
23 $\mathbf{T}^-_{\mathrm{GLR}}(i, j, k) \leftarrow \mathbf{T}(i, j, k)$
24 **end**
25 **end**
26 **end**
27 $\mathbf{T}^-_{\mathrm{GLR}} \leftarrow -\mathbf{T}^-_{\mathrm{GLR}}$
28 $\mathbf{M}^+ \leftarrow$ local maxima $(\mathbf{T}^+_{\mathrm{GLR}})$
29 $\mathbf{M}^- \leftarrow$ local maxima $(\mathbf{T}^-_{\mathrm{GLR}})$

where the superscript $^+$ refers to positive (emission) lines and $\overline{\mathbf{d}}_i$ denotes the spatial-spectral signature $\mathbf{d}_i$ to which the mean has been subtracted. Equation (8) can be efficiently computed for all positions (x, y, z) using Algorithm 2. The first main loop (rows 2 to 7) processes the cubes channel by channel. The second main loop (rows 8 to 26) processes the result of the first loop profile by profile, with an embedded loop processing spectrum per spectrum (rows 11 to 16). Comparisons of the score obtained for each profile (rows 19 to 25) implement the max and min operators required for T^+_{GLR} and T^-_{GLR} (cf. Eqs. (8) above and (10) below).

3.2.2. Local extrema

The GLR test statistics result in a cube of test statistics values, $\mathbf{T}^+_{\mathrm{GLR}}$. When a line emission is present at a position $p_0 := (x_0, y_0, z_0)$, the values of $\mathbf{T}^+_{\mathrm{GLR}}$ tend to increase statistically in the vicinity of p_0 with a local maximum at (or near) p_0. For this reason, detection approaches based on local maxima are often advocated, possibly after a matched-filtering step, when "blobs" of moderate extensions have to be detected in random fields (e.g., Sobey 1992; Bardeen et al. 1986; Vio & Andreani 2016;

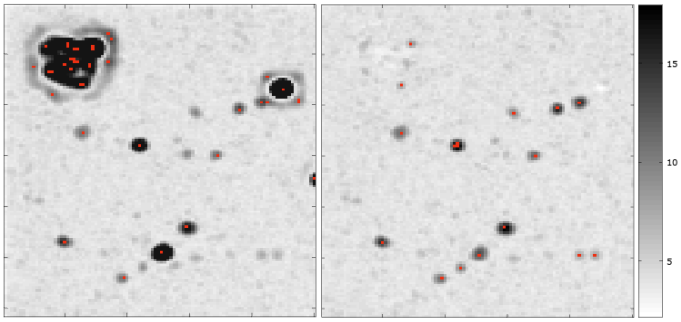

Fig. 5. Comparison of two different nuisance removal algorithms on the 100×100 top left region of the udf-10 field. *Left*: median filtering with a window of 144 channels. *Right*: DCT + PCA. Red: local maxima $\mathbf{M}^+$ larger than $v = 3\sigma$ after the mean ($v = 9.4$ for the median and $v = 6.6$ for the DCT+PCA). The grayscale ranges from 0 (white) to 16 (black).

Vio et al. 2017; Schwartzman & Telschow 2018). We opt for this approach and consider in the following the cube obtained by computing the three-dimensional local maxima of $\mathbf{T}^+_{\mathrm{GLR}}$, noted $\mathbf{M}^+$ and defined by:

$$\mathbf{M}^+(x,y,z) := \begin{cases} \mathbf{T}^+_{\mathrm{GLR}}(x,y,z) & \text{if } \mathbf{T}^+_{\mathrm{GLR}}(x,y,z) > \mathbf{v}(i), \forall i = 1\ldots,26, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where $\mathbf{v}$ collects the 26 voxels connected by faces, edges or corners touch to voxel $\mathbf{T}^+_{\mathrm{GLR}}(x,y,z)$.

An example of the resulting cube of test statistics is represented in Fig. 5, right panel. This panel shows in gray scale the maximum over the spectral index of $\mathbf{T}^+_{\mathrm{GLR}}$ obtained for a region of udf-10. The corresponding local maxima (nonzero values of $\mathbf{M}^+$) are shown in red. In this panel, $\mathbf{T}^+_{\mathrm{GLR}}$ and $\mathbf{M}^+$ reveal very clearly regions where spatially unresolved emission lines are likely to be present (darker blobs of the size of the PSF, and red points). In comparison, the left panel shows the GLR cube obtained when applying Algorithm 2 after a preprocessing based solely on a median filtering (instead of the DCT+PCA processing described above). Clearly, a less efficient nuisance removal leads, in the vicinity of bright sources for instance, to wild and undesired variations of the test statistics. Such a loss of efficiency in the preprocessing stage would have to be compensated for by a possibly time consuming postprocessing stage.

In order to evaluate how significant are the claimed detections, an important question is to know the distribution of the local maxima under $\mathcal{H}_0$. This problem has been studied in applications like the heights of waves in stationary sea state (Sobey 1992), the fluctuations of the cosmic background (Bardeen et al. 1986), source detection in radiointerferometric maps (Vio & Andreani 2016; Vio et al. 2017) or brain activity regions in neuroimaging (Schwartzman & Telschow 2018), the random fields in which local maxima are considered being two-dimensional for the former three applications and three-dimensional for the latter.

In all these works, the distribution of the local maxima are derived from theoretical results regarding Gaussian random fields, see in particular the works of Cheng & Schwartzman (2018) for recent results and a review. We opted here for a different approach however, because the random field $\mathbf{T}^+_{\mathrm{GLR}}(x,y,z)$ is not guaranteed to be Gaussian nor smooth (owing for instance to the maximum over the spectral profiles computed in (8)). Besides, it is not isotropic and the correlation structure is likely to be not stationary in the spatial dimension owing to the varying number of exposures and to telluric spectral lines. While it might be possible to derive an accurate statistical characterization of the local maxima of MUSE data by combining results of Schwartzman & Telschow (2018) for 3D, non isotropic non-stationary and possibly non Gaussian random fields, this approach would deserve a far more involved study. We present and validate here a relatively more simple and empirical approach. This approach combines inference based on local maxima with an estimation of the distribution under $\mathcal{H}_0$ directly from the data itself, as considered for instance in Walter et al. (2016) and Bacher et al. (2017) and further motivated here by the large dimension of the data cubes. For this, our approach is in two steps.

– Step 1: Compute GLR scores for absorption lines under $\mathcal{H}_0$. Consider that we wish to detect absorption (rather than emission) lines. The model is the same as (7) but with this time $b < 0$. The GLR leads to the tests statistics

$$T^-_{\mathrm{GLR}} := -\min_i \left(0, \frac{-\mathbf{f}^\top \bar{\mathbf{d}}_i}{\|\bar{\mathbf{d}}_i\|}\right). \tag{10}$$

As shown in Appendix A.2, under $\mathcal{H}_0$, T^+_{GLR} and T^-_{GLR} share the same distribution.

– Step 2 : Statistics of the local maxima. Since the distribution of T^+_{GLR} and T^-_{GLR} are the same, a training set of test statistics for the local maxima $\mathbf{M}^+$ can be obtained by the local maxima of T^-_{GLR}, noted $\mathbf{M}^-$, which is defined similarly as $\mathbf{M}^+$ in Eq. (9), with $\mathbf{T}^-$ replacing $\mathbf{T}^+$. In practice, the background region in which the statistics are estimated is obtained by the merged segmentation maps obtained from tests t_1 and t_2 in Eq. (5).

3.3. Purity estimation

The purity (or fidelity) is defined as the proportion of true discoveries (called S below) with respect to the total number of discoveries (R). With these notations the number $V := R - S$ is the number of false discoveries and the purity is

$$P := \frac{S}{R} = \frac{R-V}{R} = 1 - \frac{V}{R}, \tag{11}$$

where $\frac{V}{R}$ is sometimes called False Discovery Proportion (FDP). We note that when a large number of comparisons is made in multiple testing procedures (as this is the case here with millions of local maxima), controlling the false alarm (or familywise error) rate at low levels leads to drastically increase the test's threshold, which results in a substantial loss of power. In such situations it can be more efficient to control instead the False Discovery Rate (FDR, Benjamini & Hochberg 1995), defined as

$$\mathrm{FDR} := \mathrm{E}\,(\mathrm{FDP}) = 1 - \mathrm{E}(P), \tag{12}$$

where $E(\cdot)$ denotes expectation and by convention FDR $= 0$ if $R = 0$. Definition (12) shows that procedures aimed at controlling purity and FDR are indeed connected.

Coming back to our line detection problem, we wish to find a procedure making it possible to decide which local maxima of $\mathbf{M}^+$ should be selected so that the purity of the resulting selection is guaranteed at a level prescribed by the user. This amounts to find the correspondence between a range of thresholds and the resulting purity.

As mentioned above, our choice is to estimate the number of discoveries V from the data itself, namely from the statistics of the local maxima $\mathbf{M}^-$, since their distribution is the same as that of $\mathbf{M}^+$ when only noise is present. Let us denote by N^- the number of voxels found in the background region[3] and by $V^-(\gamma)$ the

[3] For udf-10, $N^- \approx 2.4 \times 10^8$, which represents ≈64% of the total number of voxels N ($N \approx 3.8 \times 10^8$).

number of local maxima of $\mathbf{M}^-$ with values larger than a given threshold $\gamma > 0$ in that region. If $\mathbf{M}^+$ was obtained from a noise process with the same statistics as $\mathbf{M}^-$, an estimate of the number of false discoveries $\widehat{V}$ that would be obtained by thresholding the full data cube $\mathbf{M}^+$ (entailing N voxels) at a value γ is

$$\widehat{V}(\gamma) = \frac{N}{N^-} V^-(\gamma). \tag{13}$$

Hence, if we denote by $R(\gamma)$ the number of local maxima above the threshold in $\mathbf{M}^+$, the purity can be estimated as

$$\widehat{P}(\gamma) := 1 - \frac{\widehat{V}(\gamma)}{R(\gamma)} = 1 - \frac{N}{N^-} \cdot \frac{V^-(\gamma)}{R(\gamma)}. \tag{14}$$

This approach is very similar to the Benjamini & Hochberg procedure for controlling the FDR. The difference is that the probabilities of local maxima to be larger than some value under $\mathcal{H}_0$ (the P-values) are estimated directly from $\mathbf{M}^-$ instead of relying on a theoretical distribution of the local maxima. Figure 6 shows that this procedure allows for an efficient control of the purity. The value of the threshold γ^* such that $\widehat{P}(\gamma^*) = P^*$ (with P^* the purity chosen by the user) is selected and $\mathbf{M}^+$ is thresholded at this value.

3.4. Pre-detection of bright emission lines

The nuisance removal via the PCA described in Sect. 3.1.3 is aimed at removing any source that is not a faint unresolved emission line. The counterpart of the efficiency of Algorithm 1 is that powerful emission lines are detected by tests t_2 in Algorithm 1, so their contribution is included in matrix $\mathbf{N}$, captured by the PCA and thus removed from the data cube. It is thus necessary to make a predetection of bright emission lines. Such lines are easily detectable by a high peak emission in the residual data cube $\widetilde{\boldsymbol{R}}$. The detection procedure for bright emission lines mirrors that described in Sect. 3.3, simply replacing $\mathbf{M}^+$ and $\mathbf{M}^-$ by the local maxima of $\widetilde{\mathbf{R}}$ and $-\widetilde{\mathbf{R}}$. For udf-10, the target purity of this stage is set to $P^* = 0.95$.

3.5. Line merging and source extraction

Once the detections are available, we need to group them into sources, where a given source can have multiple lines. This can be a tricky step in regions where bright continuum sources are present because such sources can lead to detections at different spatial positions despite the DCT+PCA (see Fig. 5). To solve this problem we use the information from a segmentation map (that can be provided or computed automatically on the continuum image to identify the regions of bright or extended sources, cf. Sect. 3.1.2) and we adopt a specific method for detections that are in these areas.

First, the detections are merged based on a spatial distance criteria (parameter called `tol_spat`). Starting from a given detection, the detections within a distance of `tol_spat` are merged. Then by looking iteratively at the neighbors of the merged detections, these neighbors are merged in the group if their distance to the seed detection is less than `tol_spat` voxels, or if the distance on the wavelength axis is less than a second parameter, `tol_spec` (that is, if a line is detected almost at the same wavelength but a different spatial position). This process is repeated for all detections that are not yet merged.

Then we take all the detections that belong to a given region of the segmentation map, and if there is more than one group of lines from the previous step we compute the distances on the wavelength axis between the groups of lines. If the minimum distance in wavelength is less than `tol_spec`, the groups are merged.

Finally, for each line we then compute a detection image, obtained by summing the GLR datacube on a window centered on the detection peak wavelength and a width of $2 \times FWHM$, where FWHM is the width of the spectral template that provides the highest correlation peak. We also compute the corresponding spectrum by weighted summation over the spatial extent of the line using the detection image as weight.

4. Application to MUSE HUDF datacube

ORIGIN was initially developed for the blind search of Lyα emitters in the MUSE deep exposure of the *Hubble* Ultra Deep Field (HUDF). A preliminary version of the code was successfully used for the blind search exploration of the entire HUDF field of view (Bacon et al. 2017). It resulted in the detection of 692Lyα emitters (Inami 2017), including 72 not detected in the HST deep broad band images (Bacon et al. 2017; Maseda et al. 2018).

Here we use the latest version of the udf-10 MUSE datacube (see Fig. 1 of Bacon et al. 2017), the single 1 arcmin^2 datacube that achieves a depth of 30 hours. In this field Inami (2017) report the detection of 158Lyα emitters, including 30 not detected in the HST deep broad band images. Compared to the previous version datacube of the same field used in Bacon et al. (2017), the data benefits from an improved data reduction pipeline, resulting in less systematics. This version is the one used in the MUSE HUDF data release II (Bacon et al., in prep.). In this section we focus on the use and performance of the algorithm with an emphasis on the Lyα emitters search.

4.1. Processing

The released version 1.0 of ORIGIN was used on the MUSE udf-10 datacube with the parameters as given in Appendix C. While most of the parameters can be used with their default value, a few need more attention: the probability of false alarm for the PCA ($P_{\mathrm{FA}}^{\mathrm{PCA}}$, Sect. 3.1.3) and the allowed minimum purity (P^*, Sect. 3.3). A correct setting of the first parameter ensures that the signal coming from bright continuum sources and the systematics left by the data reduction pipeline are properly removed. When $P_{\mathrm{FA}}^{\mathrm{PCA}}$ is too low the test threshold is too high and too few spectra are cleaned, leaving nuisances in the residual data cube. When $P_{\mathrm{FA}}^{\mathrm{PCA}}$ is too large the test threshold is too low and the signal from some emitters can be impacted. After some trials, a value of 10% was used (see an example in Fig. 3). Note that with such a value, 7% of the emitters (the brightest ones) are killed by the PCA process, but they are recovered in the last step of the method (Sect. 3.4).

The second parameter, the target purity P^*, impacts the detection threshold above which the local maxima of the test statistics are considered as detected lines. A purity value of 80% was selected as a compromise between the number of false and true detections. The impact and the reliability of this parameter on the detection performance is discussed in later in Sect. 4.3.

We also spent some time to the design of an "optimum" input spectral line dictionary (Sect. 3.2). After some trials we found that a set of 3 Gaussian profiles with FWHM of 2, 7 and 12 spectral pixels offers both good detection power (completeness) and affordable computational load. A lower number of profiles degrades the detection power while a higher number does not increase it but requires more computing power. More

sophisticated profiles (e.g., asymmetric line shape mimicking those generally found in Lyα emitters) were also considered but not used given their negligible impact on the performances.

4.2. Results

The algorithm detects 791 emission lines belonging to 446 different sources. The algorithm assigns to each detected emission line a significance level, computed as the maximum purity at which this line can be detected (hence, the significance level of all detected lines is above P^*, which is 80% here).

After careful inspection by a set of experts, we confirm 248 Lyα emitters covering a broad range of redshifts (2.8–6.7) and 133 lower redshifts galaxies, mostly [O II] emitters but also a few nearby Hα emitters[4]. Table 1 gives the success rate of redshift assignment for the full sample as a function of the purity estimation. Note that the measured success rate is smaller than the expected purity. This apparent discrepancy is due to the fundamental difference between a line detection and a redshift assignment. The latter is a very complex process involving matching spectral signature with template, searching for multiple lines, measuring line shape and even performing deblending of the source from its environment. Thus at low S/N it is not surprising that some of the real detections do not lead to a confirmed redshift.

A few representative examples of ORIGIN detections are given in Fig. 7. In the following we discuss each case.

The first case (ORIGIN ID 310, first row of Fig. 7) shows the detection of a Lyα emitter. The source cannot be seen in the MUSE white light image but is clearly visible in the HST deep *F775W* image. The corresponding ORIGIN detection peaked at 7459 Å. The pseudo narrow band detection image is obtained by averaging the datacube of the GLR test statistics as described in Sect. 3.2, on a window centered on the detection peak (see Sect. 3.5). A similar image can be obtained by performing the same operation on the raw datacube. As expected the corresponding narrow-band image is more noisy compared to the detection image. However, the smoothed narrow band image displays a clear signal, in line with ORIGIN detection. The corresponding bright emission line seen in the source spectrum is broad and asymmetric with a tail on the red side of the peak, a characteristic of Lyα emission line. The redshift of the source is 5.13. The HST matched source has the ID 10185 in the Rafelski catalog (Rafelski et al. 2016), its magnitude is AB 28.8 in *F775W* and its Bayesian photometric redshift is 0.61, but with a large error bar (0.47–4.27). For such a faint source the photometric redshift accuracy is not much reliable, even in this field which has an exquisite suite of deep images spanning a large wavelength range from UV to IR (Brinchmann 2017). This demonstrates the power of a blind detection method like ORIGIN, which does not rely on prior information.

The second case (ORIGIN ID 204) shows the detection of a bright [O II] emitter at $z = 1.3$. The source is bright (*F775W* AB 24.5) and clearly visible in MUSE white light and HST broad band images. The [O II] doublet is prominent in the spectrum. While the reconstructed narrow-band image displays a well defined spatial structure that looks like the broad band image, this is not the case of the detection image, which displays a hole at the location of the galaxy center. This is due to the PCA continuum subtraction (Sect. 3.1.3), which has removed the brightest [O II] emission, leaving only the faintest [O II] lines in the outskirts of the galaxy. In this case the [O II] emission was recovered in the predetection stage in the S/N datacube (see Sect. 3.4).

The third source (ORIGIN ID 253) is a good example of the power of ORIGIN blind detection in the vicinity of a bright continuum object. ORIGIN has detected an emission line in the vicinity of the nearby ($z = 0.62$) bright (AB 22) spiral galaxy ID 23794 (Rafelski et al. 2016). The detection and narrow-band images display a coherent structure that spatially matches a small source in the *F775W* image. Without the ORIGIN detection, one would have assumed that this small source in the HST image is one of the H_2 region, which can be seen along the spiral arm of the galaxy. The fact that the multi-band HST segmentation map did not identify the source as a different object from the main galaxy would have strengthened this (false) assumption. However, the spectrum confirmed that the ORIGIN line does not belong to the main galaxy, but is a Lyα emitter at $z = 4.7$ with a typical asymmetric line profile.

The next example (ORIGIN ID 334) shows the detection of an emission line in the external part of a galaxy. But contrarily to the previous case, the narrow band and detection images do not show the same structure. The narrow band image points to the galaxy core, while the detection image does not show much signal there. After inspection of the two spectra, one can demonstrate that the line detected by ORIGIN is the H10 Balmer line belonging to the main galaxy spectrum. This line was faint enough to be left by the PCA continuum subtraction. This example shows that care must be taken when analyzing detection results in the region of bright continuum sources.

The last two examples (ORIGIN ID 376 and 329) display detection of faint Lyα emitters without HST counterpart. The

Table 1. Success rate of redshift assignment for udf-10 ORIGIN detections.

Purity	Ndetect	withZ	noZ	Success rate
1.00–0.95	303	274	29	90.4%
0.95–0.90	62	28	34	82.7%
0.90–0.85	30	11	19	79.2%
0.85–0.80	51	13	38	73.1%

Notes. Ndetect corresponds to the number of sources detected in the considered purity range, withZ (resp. noZ) correspond respectively to the number of successful (resp. unsuccessful) redshift assignments.

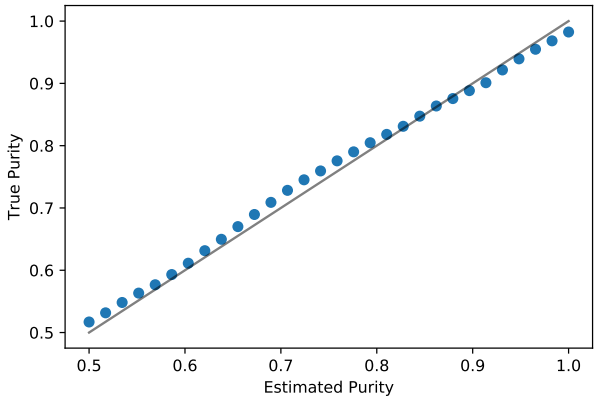

Fig. 6. Comparison of the ORIGIN estimated purity $\widehat{P}$ (cf. Eq. (14)) versus the true purity (Eq. (11)) for the udf-10 fake datacube.

[4] We also detect a few [C III] emitters, as well as [O III] emitters but only a few with respect to the [O II] emitters which are ten times more numerous. Note that most of [C III] emitters have low equivalent width and strong continuum.

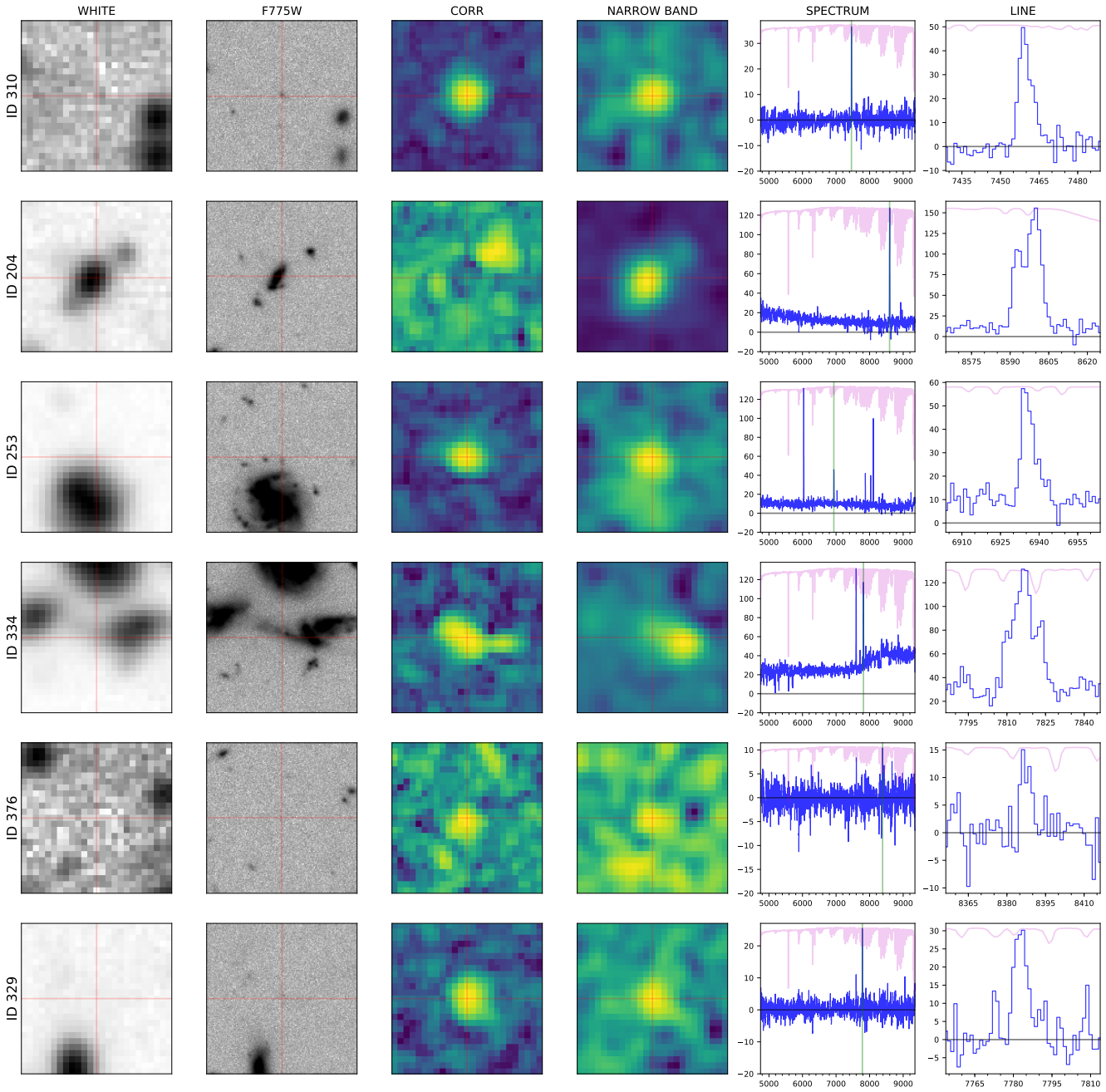

Fig. 7. Examples of ORIGIN detections in udf-10. For each detected source, we show the white light image reconstructed from the MUSE datacube (*WHITE column*), the corresponding HST image in the *F775W* filter (the *Hubble* ACS broad band filter *F775W* has an effective wavelength of 7624 Å which is nearly aligned with the central wavelength of MUSE (7000 Å). Its effective band pass of 1299 Å covers a third of the MUSE band pass (4650–9350 Å).) (*F775W column*), the detection image obtained by the averaging the GLR datacube over the detected wavelengths (*CORR column*) and the narrow band image obtained by averaging the raw datacube over the same wavelength range (*NARROW BAND column*). Note that the narrow band image is smoothed with a 0.6″ FWHM Gaussian kernel to enhance visually possible sources. The box size of the images is 5 arcsec. The source spectrum (smoothed with a boxcar of 5 pixels) is shown in the *SPECTRUM column*. The corresponding noise standard deviation is displayed in magenta (inverted on the y axis) and the green line displays the wavelength of the ORIGIN detection. The *last column* (LINE) displays the (unsmoothed) spectrum zoomed around the ORIGIN detected wavelength.

sources are clearly visible in the detection images and to some extent also in the narrow band images, but nothing is detected in any HST bands. The spectrum shape is also indicative of Lyα emission. Given the depth of the HST images (AB ∼30) in the *Hubble* UDF, a low redshift object is excluded and the most likely solution corresponds to high redshift Lyα emitters. The reality of these detections has been confirmed by Maseda et al. (2018), who have demonstrated that the Lyman-break signal can be recovered by stacking the HST broad-band images of the ORIGIN detected Lyα emitters without HST counterparts.

4.3. Purity and completeness

The purity, that is, the fraction of true detections with respect to the total number of detections, is a built-in capability of ORIGIN (see Sect. 3.3). Obviously we would like to minimize

the number of false detections in the total list of detections in order to get a purity as close as possible to 100%. On the other hand, targeting a higher purity automatically decreases the completeness, that is, the fraction of lines that are detected by the algorithm with respect to the total number of sources to be discovered. The trade-off between purity and completeness is a feature of any detection method.

The estimation of completeness is highly dependent of the sources we want to detect. For example, the completeness is different for a population of unresolved bright Lyα emitters , or for a population of unresolved faint Lyα emitters or for a population of diffuse emission sources with broad emission lines. Hence, a generic estimation of completeness is not a built-in function of ORIGIN (in contrast to the estimation of the purity) neither of any detection method. A detailed study of Lyα emitters completeness in the HUDF datacubes is beyond the scope of this paper and will be presented in the upcoming DR2 paper (Bacon et al., in prep.). Nevertheless, we address here the question of completeness with a simpler approach by generating fake Lyα emitters into a datacube with similar characteristics to the HUDF datacube.

In practice we replace the signal of the udf-10 datacube by a random Gaussian noise with zero mean and variance equal to the udf-10 variance estimate. We then generate fake Lyα emitters using typical number counts representative of the faint end Lyα luminosity function (Drake 2017). The generated Lyα lines are asymmetric with FWHM and skewness that are representative of the Lyα emitters population. The resulting spectral profiles of the Lyα emitters, supposed to be point sources, are convolved with the MUSE PSF. Finally, we add 9 bright continuum sources. The resulting data cube is an idealized version of the real data cube but with the same noise characteristics. Note that this process assumes a Gaussian noise distribution, an assumption checked by Bacon et al. (2017) in the udf-10 datacube.

ORIGIN is then run on the fake datacube, varying the fluxes, the locations and the wavelengths of the fake Lyα emitters. The comparison of the ORIGIN detected sources with the input list allows to compute the true purity and completeness.

Figure 6 compares the purity estimated by the algorithm with respect to the true purity. This shows that the estimate provided by ORIGIN is reliable on a wide range of purity levels. Figure 8 shows completeness versus purity plots for two different wavelength ranges and three flux values. As expected bright sources are fully recovered while the algorithm achieves a lower completeness for fainter ones. Note also that completeness is lower in the red for a given flux because of the impact of sky lines (the larger variance in the red end is visible in the magenta plots of Fig. 7, column SPECTRUM). Finally, note that the weak slopes of the completeness versus purity curves indicate that ORIGIN is able to achieve a relatively high completeness (with respect to the "asymptotic" one) with a fairly high purity (0.8 for instance).

4.4. Discussion

With respect to the preliminary, unreleased version of ORIGIN used on the version 0.42 of the udf-10 datacube in Bacon et al. (2017), we have increased the number of detections of Lyα emitters by 34% (255 versus 190). Note that the difference is only 7% (174 versus 163) when we restrict the sample to the highest confidence redshifts. This was expected as the improved performance of ORIGIN has allowed the detection of fainter Lyα emitters while the "easiest one" were already identified in the preliminary version of the algorithm.

We also report the detection of 86Lyα emitters without HST counterpart. This is almost 3 times more sources with respect

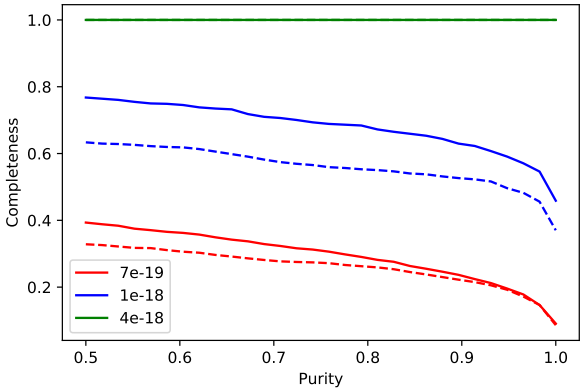

Fig. 8. Estimation of completeness versus purity for the udf-10 fake datacube. The values are given for three fluxes (in cgs units) and two wavelength ranges: 6000–7000 Å (solid lines) and 8000–9000 Å (dashed lines).

to the 30 sources published in Bacon et al. (2017). The ability to detect such sources, which are invisible in the deepest HST broad band images, is an important feature of ORIGIN.

While some of these improvements are due to the better data quality of the latest version of the datacube, most of them result from the advanced methodological features of the present version of ORIGIN with respect to the preliminary version used in Bacon et al. (2017). The most important improvements regard the continuum subtraction and the use of robust test statistics. Last, but not least, ORIGIN gives now an important reliability factor, the estimated purity, a must for a robust detection method.

Although the DCT and iterative PCA do a good job for the continuum subtraction (in particular better than a basic median filtering, see Sect. 3.1.3), it still leaves some low level residuals that can produce spurious detections (such an example – ORIGIN ID 334– is highlighted in Sect. 4.2). The consequence is that the purity in the regions corresponding to bright continuum sources is lower than in the background region. A precise estimate of the purity in that case is difficult.

Among the many results brought by the analysis of the MUSE deep field observations is the ubiquitous presence of Lyα low surface brightness extended halos surrounding Lyα emitters (Wisotzki et al. 2016, 2018; Leclercq 2017). Given that ORIGIN was designed for point source detection one can ask whether a better strategy would have been to use a kernel larger than the PSF. Even if the total flux of the Lyα halo can be similar to or even larger than the flux of the central component, its surface brightness is approximately two magnitude lower than the central component outside the PSF area (see Fig. 2 of Leclercq 2017). Hence, in practice, the point like assumption does not appear to be an important limiting factor for the blind detection of Lyα emitters .

5. Summary and conclusion

The algorithm ORIGIN is designed to be as powerful as possible for detecting faint spatial-spectral emission signatures, to allow for a stable false detection rate over the data cube and to provide an automated and reliable estimation of the resulting purity. We have shown on realistic simulated MUSE data that the algorithm achieves these goals and ORIGIN was applied to the deepest MUSE observations of the *Hubble* Ultra Deep Field (udf-10). In this tiny $1' \times 1'$ field, ORIGIN revealed a large population of Lyα emitters (255), including 86 sources

without HST counterpart. These sources could not have been found without such a powerful blind detection method as ORIGIN.

While the algorithm is mostly automated, we have presented a list of the main parameters with guidelines allowing to tune their default values. The algorithm is freely available to the community as a Python package on GitHub.

We have already identified points for improvements to the current version of ORIGIN and we are currently working on them. The first point is the greedy PCA (Sect. 3.1.3). Although this step works quite well and is fast, it may still leave residual nuisances that increase the false detection rate in the region of bright extended sources. Besides, we have not analyzed the behavior of this step in different acquisition settings, for instance crowded fields or fields with much lower signal-to-noise ratios than udf-10. Finally, we are developing a method for automatically tuning of the $P_{\mathrm{FA}}^{\mathrm{PCA}}$ parameter involved in this stage.

We also know that there is room for improvement in the detection power (and completeness) of the method, by accounting for a more complex model than model (7) (leading then to GLR test statistics different from (8)). We have already devised slightly more powerful models and tests' statistics, but those make the total amount of computing power far too demanding, as they impose a processing subcube per subcube (instead of allowing for fast convolutions along the spatial and spectral dimensions). Improvements might yet be found in this direction in the future.

Finally, the estimation of the purity can also be improved. While the current estimation is efficient for data cubes with extended zones of "background", we know that the procedure may fail for crowded data cubes where no or too few such zones exist. In such cases, the amount of data available to estimate the test statistics under the null hypothesis is insufficient, a problem which is the object of active research in statistics.

Acknowledgements. DM made extensive use of the free software GNU Octave (Eaton et al. 2018) for developping ORIGIN and is in particular thankful to the developers of the Octave packages *statistics*, *signal* and *image*. DM also acknowledges support from the GDR ISIS through the *Projets exploratoires* program (project TASTY). Part of this work was granted access to the HPC and visualization resources of the Centre de Calcul Interactif hosted by University of Nice Côte d'Azur. RB acknowledges support from the ERC advanced grant 339659-MUSICOS. This research made use of the following open-source packages for Python and we are thankful to the developers of these: Astropy (Astropy Collaboration et al. 2013, 2018), Matplotlib (Hunter 2007), MPDAF (Piqueras et al. 2017), Numpy (van der Walt et al. 2011), Photutils (Bradley et al. 2019), Scipy (Jones et al. 2001).

References

Astropy Collaboration (Robitaille, T. P., et al.) 2013, A&A, 558, A33
Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, AJ, 156, 123
Bacher, R., Meillier, C., Chatelain, F., & Michel, O. J. J. 2017, IEEE Trans. Signal Process., 65, 3538
Bacon, R., Accardo, M., Adjali, L., et al. 2010, SPIE Conf. Ser., 7735, 8
Bacon, R., Vernet, J., Borosiva, E., et al. 2014, The Messenger, 157, 21
Bacon, R., Brinchmann, J., Richard, J., et al. 2015, A&A, 575, A75
Bacon, R., Conseil, S., Mary, D., et al. 2017, A&A, 608, A1
Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, ApJ, 304, 15
Benjamini, Y., & Hochberg, Y. 1995, J. R. Stat. Soc. Ser. B (Method.), 57, 289
Bradley, L., Sipocz, B., Robitaille, T., et al. 2019, https://doi.org/10.5281/zenodo.2533376
Brinchmann, J., Inami, H., Bacon, R., et al. 2017, A&A, 608, A3
Cheng, D., & Schwartzman, A. 2018, Bernoulli, 24, 3422
Drake, A. B., Guiderdoni, B., Blaizot, J., et al. 2017, MNRAS, 471, 267
Eaton, J. W., Bateman, D., Hauberg, S., & Wehbring, R. 2018, GNU Octave version 4.4.0 Manual: a High-level Interactive Language for Numerical Computations
Herenz, C. H., & Wisotzki, L. 2017, A&A, 602, A111
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Inami, H., Bacon, R., Brinchmann, J., et al. 2017, A&A, 608, A2
Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python
Kay, S. M. 1998, Fundamentals of Statistical Signal Processing: Detection Theory, Vol. 2 (Prentice-Hall PTR)
Lagattuta, D. J., Richard, J., Bauer, F. E., et al. 2019, MNRAS, 485, 3738
Leclercq, F., Bacon, R., Wisotzki, L., et al. 2017, A&A, 608, A8
Maseda, M. V., Bacon, R., Franx, M., et al. 2018, ApJ, 865, L1
Meillier, C., Chatelain, F., Michel, O., et al. 2016, A&A, 588, A140
Paris, S., Suleiman, R. F. R., Mary, D., & Ferrari, A. 2013, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (IEEE), 3947
Piqueras, L., Conseil, S., Shepherd, M., et al. 2017, ADASS XXVI proc., in press [arXiv:1710.03554]
Rafelski, M., Gardner, J. P., Fumagalli, M., et al. 2016, ApJ, 825, 87
Scharf, L. L., & Friedlander, B. 1994, IEEE Trans. on Signal Processing, 42, 2146
Schwartzman, A., & Telschow, F. 2018, bioRxiv [https://www.biorxiv.org/content/early/2018/06/28/358051.full.pdf]
Sobey, R. 1992, Ocean Eng., 19, 101
Suleiman, R., Mary, D., & Ferrari, A. 2013, Proc. ICASSP, 2013
Suleiman, R., Mary, D., & Ferrari, A. 2014, IEEE Trans. Signal Process., 62, 5973
Urrutia, T., Wisotzki, L., Kerutt, J., et al. 2019, A&A, 624, A141
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Comput. Sci. Eng., 13, 22
Vio, R., & Andreani, P. 2016, A&A, 589, A20
Vio, R., Vergès, C., & Andreani, P. 2017, A&A, 604, A115
Walter, F., Decarli, R., Aravena, M., et al. 2016, ApJ, 833, 67
Wisotzki, L., Bacon, R., Blaizot, J., et al. 2016, A&A, 587, A98
Wisotzki, L., Bacon, R., Brinchmann, J., et al. 2018, Nature, 562, 229

Appendix A: Mathematical description

A.1. Derivation of T^+_{GLR}

The Generalized Likelihood Ratio computes the ratio of the likelihoods under both hypotheses, with the unknown parameters set as their maximum likelihood estimates. For the composite model (7) this leads to

$$\text{GLR:} \quad \frac{\max_{i,a,b} p(\mathbf{f}; i, a, b)}{\max_{a} p((\mathbf{f}; a)}. \tag{A.1}$$

For i fixed, this problem has been considered by Scharf & Friedlander (1994). From their expression (5.15), where $\boldsymbol{x}, \mathbf{S}$ and $\mathbf{y}$ correspond respectively to $\mathbf{d}_i$, $\mathbf{1}$ and $\mathbf{f}$, the GLR test statistic is

$$T_i := \max\left(0, \frac{\mathbf{f}^\top \mathrm{P}^\perp_{\mathbf{1}} \mathbf{d}_i}{(\mathbf{d}_i^\top \mathrm{P}^\perp_{\mathbf{1}} \mathbf{d}_i)^{1/2}}\right), \tag{A.2}$$

where $\mathrm{P}^\perp_{\mathbf{1}}$ denotes the projection on the orthogonal complement of $\mathbf{1}$:

$$\mathrm{P}^\perp_{\mathbf{1}} := \mathbf{I} - \mathbf{1}(\mathbf{1}^\top\mathbf{1})^{-1}\mathbf{1}^\top = \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n_f}, \tag{A.3}$$

with $n_f = n_x n_y n_z$ (the number of voxels in $\mathbf{f}$). Now note that

$$\begin{cases} \mathrm{P}^\perp_{\mathbf{1}} \mathbf{d}_i = \mathbf{d}_i - \frac{\mathbf{1}\mathbf{1}^\top}{n_f}\mathbf{d}_i = \mathbf{d}_i - \frac{1}{n_f}(\mathbf{d}_i^\top \mathbf{1})\mathbf{1} := \overline{\mathbf{d}}_i, \\ \mathbf{f}^\top \mathrm{P}^\perp_{\mathbf{1}} \mathbf{d}_i = \mathbf{f}^\top \overline{\mathbf{d}}_i. \end{cases} \tag{A.4}$$

Hence, the GLR test statistic for fixed i (A.2) can be rewritten as

$$T_i = \max\left(0, \frac{\mathbf{f}^\top \overline{\boldsymbol{d}}_i}{\|\overline{\boldsymbol{d}}_i\|}\right), \tag{A.5}$$

and computing the maximum over the index i as in (A.1) leads to

$$T^+_{\mathrm{GLR}} = \max_i T_i = \max_i \left(0, \frac{\mathbf{f}^\top \overline{\boldsymbol{d}}_i}{\|\overline{\boldsymbol{d}}_i\|}\right). \tag{A.6}$$

A.2. Statistics of T^-_{GLR}

The statistics of T^-_{GLR} can be derived using an approach similar to the Proposition II.1 of Bacher et al. (2017), which we report and adapt here as our setting is slightly different in the considered model (7) and test statistics (8). First, as Bacher et al. (2017) we make the hypothesis that under $\mathcal{H}_0$ the noise is symmetric. We do not require the noise to be centred as the considered GLR tests statistics is invariant to an arbitrary shift. Second, note that for any subcube $\mathbf{f}$,

$$\frac{\mathbf{f}^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|} = -\frac{(-\mathbf{f})^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|}. \tag{A.7}$$

In words, the value of the amplitude estimated when fitting a line profile to a spectrum is the opposite of the value obtained when fitting the profile to the opposite spectrum. Third, note that for any finite set of real numbers $\{a_i\}$,

$$\max_i (0, a_i) = -\min_i (0, -a_i). \tag{A.8}$$

Consider now T^-_{GLR} in (10). We have

$$\max_i \left(0, \frac{\mathbf{f}^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|}\right) = \max_i \left(0, -\frac{-\mathbf{f}^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|}\right) \tag{A.9}$$

$$= -\min_i \left(0, \frac{-\mathbf{f}^\top \overline{\mathbf{d}}_i}{\|\overline{\mathbf{d}}_i\|}\right) \tag{A.10}$$

where the first equality comes from (A.7) and the second from (A.8). Since under $\mathcal{H}_0$, $\mathbf{f}$ and $-\mathbf{f}$ have the same distribution, the second equality above shows that T^+_{GLR} and T^-_{GLR} also share the same distribution.

Appendix B: Implementation

ORIGIN was developped in GNU Octave with a twin version ported and optimized in Python as the Python package `muse_origin`. Its source code is available on GitHub[5] under a MIT License. A complete documentation is also available on readthedocs[6].

As the processing of a MUSE data cube with the ORIGIN algorithm is relatively complex, it is divided in steps corresponding roughly to the steps described in Sect. 3. Each step produces intermediate results. This allows to stop at a given point and to inspect the results. It is possible to save the outputs after each step and to reload a session to continue the processing.

To run ORIGIN, it is first necessary to instantiate a `muse_origin.ORIGIN` object. In the considered framework this object is a MUSE data cube, which usually contains information about the FSF (if not, this information must be provided separately). The name given to this object is the `session` name used as the directory name in which outputs are saved (by default this is inside the current directory but this can be overridden with the `path` argument).

Here is an example:

```
>>> from muse_origin import ORIGIN
>>> orig = ORIGIN(CUBE, name='origtest')
INFO : Step 00 - Initialization (ORIGIN v3.2)
INFO : Read the Data Cube minicube.fits
INFO : Compute FSFs from the datacube
INFO : mean FWHM of the FSFs = 3.32 pixels
INFO : 00 Done
```

The processing steps described in this article can the be run on this object. For instance for the first step this yields:

```
>>> orig.set_loglevel('INFO')
>>> orig.step01_preprocessing()
INFO : Step 01 - Preprocessing
INFO : DCT computation
INFO : Data standardizing
INFO : Std signal saved in self.cube_std ...
INFO : Compute local maximum of std cube values
INFO : Save self.cube_local_max ...
INFO : DCT continuum saved in self.cont_dct...
INFO : Segmentation based on the continuum
INFO : Found 11 regions, threshold=1.94
INFO : Segmentation based on the residual
INFO : Found 3 regions, threshold=1.12
INFO : Merging both maps
INFO : Segmap saved in self.segmap_merged ...
INFO : 01 Done - 2.50 sec.
```

[5] https://github.com/musevlt/origin
[6] https://muse-origin.readthedocs.io/en/latest/

The detailed contents of each step are described in more details in the documentation[7], which also contains an Jupyter notebook example.

The other steps can be run with the following commands:

```
>>> orig.step02_areas()
>>> orig.step03_compute_PCA_threshold()
>>> orig.step04_compute_greedy_PCA()
>>> orig.step05_compute_TGLR()
>>> orig.step06_compute_purity_threshold()
>>> orig.step07_detection()
>>> orig.step08_compute_spectra()
>>> orig.step09_clean_results()
>>> orig.step10_create_masks()
>>> orig.step11_save_sources()
```

Each step produces various files, which are useful to analyze the algorithm's behavior and the influence of its parameters. The most important files are the final catalogs with the list of all the detected lines (`orig.Cat3_lines`) and sources (`orig.Cat3_sources`). The last step also creates MPDAF *Source files*[8], which gather all the information for each source (GLR cube, images, masks, spectra, cf. Sect. 3.5).

We underline that a substantial amount of time was devoted to optimize as much as possible the numerical implementation of the considered methods in order to minimize the overall computation time. Table B.1 gives the computation times for each step for the `udf10` data cube described in Sect. 4. These times were obtained on a computing machine with 80 Intel® Xeon® Gold 6148 CPU at 2.40 GHz CPUs, and using the optimized version of Numpy with the Intel® MKL. As ORIGIN uses intensively linear algebra for the iterative PCA and FFT for the profiles convolution, using the Intel® MKL with Numpy may bring significant performance boost. This is the case by default when using Anaconda or Miniconda, and it is also possible to use the Numpy package from Intel® with `pip`. Intel® also provides an `mkl-fft`[9] package with a parallelized version of the FFT.

Table B.1. Computation times for each step (minutes) for the `udf10` data cube described in Sect. 4.

Step	Execution time
`step01_preprocessing`	01:49
`step02_areas`	00:02
`step03_compute_PCA_threshold`	00:03
`step04_compute_greedy_PCA`	02:47
`step05_compute_TGLR`	05:31
`step06_compute_purity_threshold`	00:12
`step07_detection`	00:05
`step08_compute_spectra`	02:01
`step09_clean_results`	00:16
`step10_create_masks`	00:13
`step11_save_sources`	01:45
Total	14:49

Appendix C: ORIGIN parameters

Table C.1. ORIGIN parameters used for udf-10 processing.

Step in Sect.	Symbol	Python variable	Value	Description
3.1.1	N_{DCT}	dct_order	10	Order of the DCT
3.1.2	$P_{\mathrm{FA}}^{\mathrm{seg}}$	pfa	0.2	"False alarm" rate for tests (5)
3.1.2	$S_{\mathrm{min}}, S_{\mathrm{max}}$	minsize,maxsize	80,120	Min and max surface for the large segmentation patches
3.1.3	$P_{\mathrm{FA}}^{\mathrm{PCA}}$	pfa_test	0.1	"False alarm" rate for test t_2 in (5) during PCA cleaning
3.1.3	F_b	Noise_population	0.05	Fraction of spectra of **B** used to estimate background
3.3	N_s		3	Number of spectral profiles for the detection
3.2	P^*	purity	0.8	Target purity for faint emission lines
3.4	P^*	purity_std	0.95	Target purity for bright emission lines

[7] https://muse-origin.readthedocs.io/
[8] https://mpdaf.readthedocs.io/en/latest/source.html
[9] https://github.com/IntelPython/mkl_fft