A fast, reliable and easy method to detect within-species DNA contamination

Tiziano Dallavilla[1], *Giuseppe Marceddu*[2], *Arianna Casadei*[2], *Luca De Antoni*[2], *Matteo Bertelli*[1,2,3]

[1] MAGI'S LAB, Rovereto (TN), Italy; [2] MAGI Euregio, Bolzano, Italy; [3] EBTNA-LAB, Rovereto (TN), Italy

Abstract. *Background and aim:* Next generation sequencing (ngs) is becoming the standard for clinical diagnosis. Different steps of NGS, such as DNA extraction, fragmentation, library preparation and amplification, require handling of samples, making the process susceptible to contamination. In diagnostic environments, sample contamination with DNA from the same species can lead to errors in diagnosis. Here we propose a simple method to detect within-sample contamination based on analysis of the heterozygous single nucleotide polymorphisms allele ratio (AR). *Methods:* A dataset of 38000 heterozygous snps was used to estimate the ar distribution. The parameters of the reference distribution were then used to estimate the contamination probability of a sample. Validation was performed using 12 samples contaminated to different levels. *Results:* Results show that the method easily detects contamination of 20% or more. The method has a limit of detection of about 10%, threshold below which the number of false positives increases significantly. *Conclusions:* The method can be applied to any type of ngs analysis and is useful for quality control. Being fast and easy to implement makes it ideal for inclusion in NGS pipelines to improve quality control of data and make results more robust. (www.actabiomedica.it)

Key words: NGS, contamination, diagnostics, bioinformatics, quality

Introduction

Since the development of the Sanger sequencing technique (1,2), DNA sequencing has become a fundamental approach for the study and diagnosis of an increasing number of genetic diseases (3,4). With the advent of NGS, the cost of sequencing decreased sharply, while output capacity showed an enormous increase (5). While NGS brought huge advantages in term of cost and sequencing capacity, it also has some drawbacks that need to be taken into account. When used for diagnostic applications, within-species sample contamination is a potential problem. Different studies (6-8) have demonstrated that the multiple handling steps required to set up NGS experiments, as well as multiple sample processing, often lead to sample contamination. Contamination can produce an unusually large number of heterozygous SNPs with an unexpected allele ratio (AR), making the analysis susceptible to genotype misclassification and false positives (FPs). While different tools exist to verify the quality of NGS data, like fastQC or NGS QC Toolkit (9), little has been done to develop algorithms to detect within-species contamination. There are tools to detect contamination in prokaryotic genomes or microbial contamination in eukaryotic genomes (10), but unfortunately there is little or nothing for within-species contamination of human samples. Here we propose a method for detecting within-sample contamination, based on analysis of the AR of heterozygous SNPs. The AR of a heterozygous SNP in a clean sample is usually about 0.5, but in the presence of a contaminant we observe more heterozygous SNPs with unexpected AR, very different from 0.5. The idea behind the method is that the higher the number of SNPs with unexpected AR in a given sample, the higher the probability that the sample is contaminated.

Materials and methods

Library preparation and sample sequencing

For this study we used 894 different samples which were analyzed following the NGS workflow described below. The detailed procedure is described in (11). In-solution target enrichment was performed according to "Nextera Rapid Capture Enrichment Guide, September 2014" (Illumina, San Diego, CA, USA), with the exception of the quantity of Tagment DNA enzyme (5 μl in place of 15 μl specified in the protocol). Fifty nanograms of each genomic DNA were initially fragmented by Nextera enzymatic technology. Limited-cycle PCR was carried out to incorporate specific index adaptors to each sample library. PCR products were purified with beads, concentration measured with dsDNA BR assay kit on Qubit 2.0 Fluorometer System (Invitrogen, Carlsbad, CA, USA), while quality and average size of fragments assessed with DNA1000 Kit on the Agilent 2100 Bioanalyzer (Agilent Technologies, Palo Alto, CA). Five hundred nanograms of each indexed DNA library were combined into the 12-plex library pool, hybridized with target-specific biotinylated probes and captured using streptavidin magnetic beads. A second round of hybridization, capture, PCR amplification and PCR clean-up were performed. The final enriched pooled libraries were quantified using the dsDNA BR assay kit on Qubit 2.0 Fluorometer System (Invitrogen, Carlsbad, CA, USA), quality and average size of fragments, mainly distributed between 500 and 600 bp, were verified using HS DNA Kit on the Agilent 2100 Bioanalyzer (Agilent Technologies, Palo Alto, CA). The pool (12-plex library) was sequenced on a MiSeq personal sequencer (Illumina, San Diego, CA) according to the manufacturer's instructions (150 bp paired-end read sequencing, MiSeq kit V3).

Variant Selection

The reference dataset of SNPs was built from the annotated VCF files of 894 samples. The samples used for this study were filtered prior to use in DNA contamination analysis. For each sample we kept only heterozygous SNPs with a mapping quality superior to 18 and that were sequenced with a coverage of at least 10X. This procedure allowed us to discard poorly sequenced SNPs with unexpected ARs due to analytical artifacts.

Detecting contamination

Our method is based on the hypothesis that contaminated samples contain more heterozygous SNPs with unexpected AR (very different from 0.5) than non-contaminated samples. The first step consisted in determining the mean, standard deviation and 95% confidence interval (CI95) of the AR distribution of the reference dataset of SNPs. Since at this point we did not have a method to distinguish whether a sample is contaminated or not, a large sample number was necessary to mitigate the effect of any contaminated samples in the reference dataset. In our case we used 894 samples (total SNPs about 38000) to generate the distribution of AR. The mean (μ) and standard deviation (σ) of the distribution were obtained as shown in Eq.1 and Eq.2 . To calculate the probability of contamination, we proceeded as follows. The SNPs present in the VCF file of the sample under investigation were filtered as described in the Variant Selection section. We then calculated the z-score of each filtered SNP as in Eq.3 using the reference distribution parameter. The z-score reflected the number of standard deviations by which the AR of a SNP was above/below the mean of our reference dataset. The CI95 of the reference dataset indicated the expected range of values of AR in which 95% of SNPs should fall, enabling us to determine how many SNPs of the study sample fell outside this window. We counted how many SNPs had a z-score outside the range -1.96/+1.96 and we divided this number by the total number of SNPs in the sample to obtain the percentage of SNPs with unexpected AR, namely the sample score. The higher the number of SNPs with z-score outside this region, the higher the probability of the sample of being contaminated. The threshold at which we consider the sample to be contaminated can be set according to the lowest contamination that we wish to detect and the number of FPs we are willing to accept in the analysis.

$$\mu = \frac{1}{N} \cdot \sum x_i \qquad (1)$$

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N-1}} \qquad (2)$$

$$z_i = \frac{s_i - \mu}{\sigma} \qquad (3)$$

Preparing contaminated samples

Algorithm performance was tested on samples contaminated artificially at different levels. To generate the contaminated samples, we used three Coriell samples, NA20828, NA20582 and NA20763. First, we measured the concentration of each sample using a BiospecNano Spectrophotometer system (Shimadzu Corporation, Japan) and we diluted them to a concentration of 20 ng/µl. Then, we measured again the concentration of each 20 ng/µl sample using a dsDNA BR Assay Kit on a Qubit 2.0 Fluorometer System (Invitrogen, Carlsbad, CA, USA) to have the most accurate value. At the end, we diluted the samples to the final desired concentration of 5 ng/µl in 10 µl final volume according to Illumina Nextera Rapid Capture Protocol. We prepared 5, 7 and 2 aliquots of NA20828, NA20582 and NA20763 at 5 ng/µl, respectively, to have enough material to combine for the contamination process. NA20828, NA20582 were used as principal samples and were contaminated at different levels with NA20582 and NA20763 respectively. A total of nine samples with known contaminations ranging from 2% to 20% were generated (Table 1). The algorithm was tested on 12 samples, the nine contaminated plus the three Coriell samples as controls. All the final samples were at 5 ng/µl in 10 µl volume.

Table 1. Summary of artificially contaminated samples. Starting with three Coriell samples, we generated nine samples contaminated at different levels. 'Sample name' indicates the name given to the sample generated, 'Mixed samples' indicates the Coriell sample used to generate the contaminated sample, 'Contamination %' indicates the percentage of contamination, 'Volume of principal sample' and 'Volume of contaminant' indicate the proportion used to generate the contaminated sample

Sample name	Mixed samples	% of contamination	Volume of principal sample (µl)	Volume of contaminant sample (µl)
C210	NA20828+NA20582 (contaminant)	10	9	1
C27	NA20828+NA20582 (contaminant)	7	9.3	0.7
C25	NA20828+NA20582 (contaminant)	5	9.5	0.5
C22	NA20828+NA20582 (contaminant)	2	9.8	0.2
C320	NA20582+NA20763 (contaminant)	20	8	2
C310	NA20582+NA20763 (contaminant)	10	9	1
C37	NA20582+NA20763 (contaminant)	7	9.3	0.7
C35	NA20582+NA20763 (contaminant)	5	9.5	0.5
C32	NA20582+NA20763 (contaminant)	2	9.8	0.2

Software

The algorithm and all the code used was written in Python 3. The code developed for this publication is intended to be executed in jupyter notebook (12). The data structures were coded using pandas library (13,14). The kernel density estimator came from the statsmodels python library (15). The figures in this paper and in the notebook were generated with matplotlib (16). The code of the algorithm along with the data used in this paper are publicly available on github at "https://github.com/tizianoBS/dna-contamination-detector.git".

Results

Validation of our method was performed on a total of 12 samples with contamination ranging from 0 to 20% (Table 2).

Before proceeding with the analysis we wanted to verify the hypothesis that the number of SNPs with unexpected AR is higher in contaminated samples. For this purpose we first compared the distribution of AR in the reference dataset and in the contaminated samples (Fig. 1). Interestingly, we noticed that the AR distribution of non contaminated samples was normal with a low standard deviation, while in contaminated samples data tended to deviate from normality and the standard deviation was much larger than for clean samples due to imbalance in the AR of SNPs. As hypothesized, the number of SNPs in the unexpected AR region (beyond CI95 of the reference dataset) was much higher in the contaminated samples.

Table 2 shows the percentage of SNPs with a z-score outside the expected region for each sample. These results show that the method readily detects contamination around 20%, and seems to indicate a limit of detection around 10%-7%, since two of the non-contaminated samples used as control had scores around those contamination percentages.

To better investigate the limit of detection of this method we compared the score of the contaminated samples with those of the 894 samples of the reference dataset. Fig. 2 shows the score of the contaminated samples (black and red line) compared to the mean (dotted blue line) and CI95 (dotted green line) of the reference dataset scores. The plot confirms that it is

Table 2. Summary of the z-score percentages of contaminated samples and controls used in validation. The z-score % of a sample indicates the percentage of SNPs in the sample with a z-score outside the expected region of -1.96/+1.96. 'Sample' indicates the name of the sample, 'z-score %' the sample score, 'Contamination %' the percentage of contamination in the sample, 'Number of SNPs' the number of variants in the sample in the VCF file after filtering, and 'Total SNPs outside threshold' indicates how many SNPs had an unexpected z-score

Sample name	Z-score %	% of contamination	Number of SNPs	Total SNPs outside threshold
C320	39.3	20	346	136
C310	13.6	10	279	38
C210	12.6	10	294	37
C27	11.3	7	275	31
C37	9.9	7	262	26
NA20763	9.5	0	284	27
NA20828	9.2	0	271	25
C25	9.2	5	272	25
C35	8.8	5	263	23
C32	7.7	2	260	20
C22	6.9	2	274	19
NA20582	5.8	0	271	15

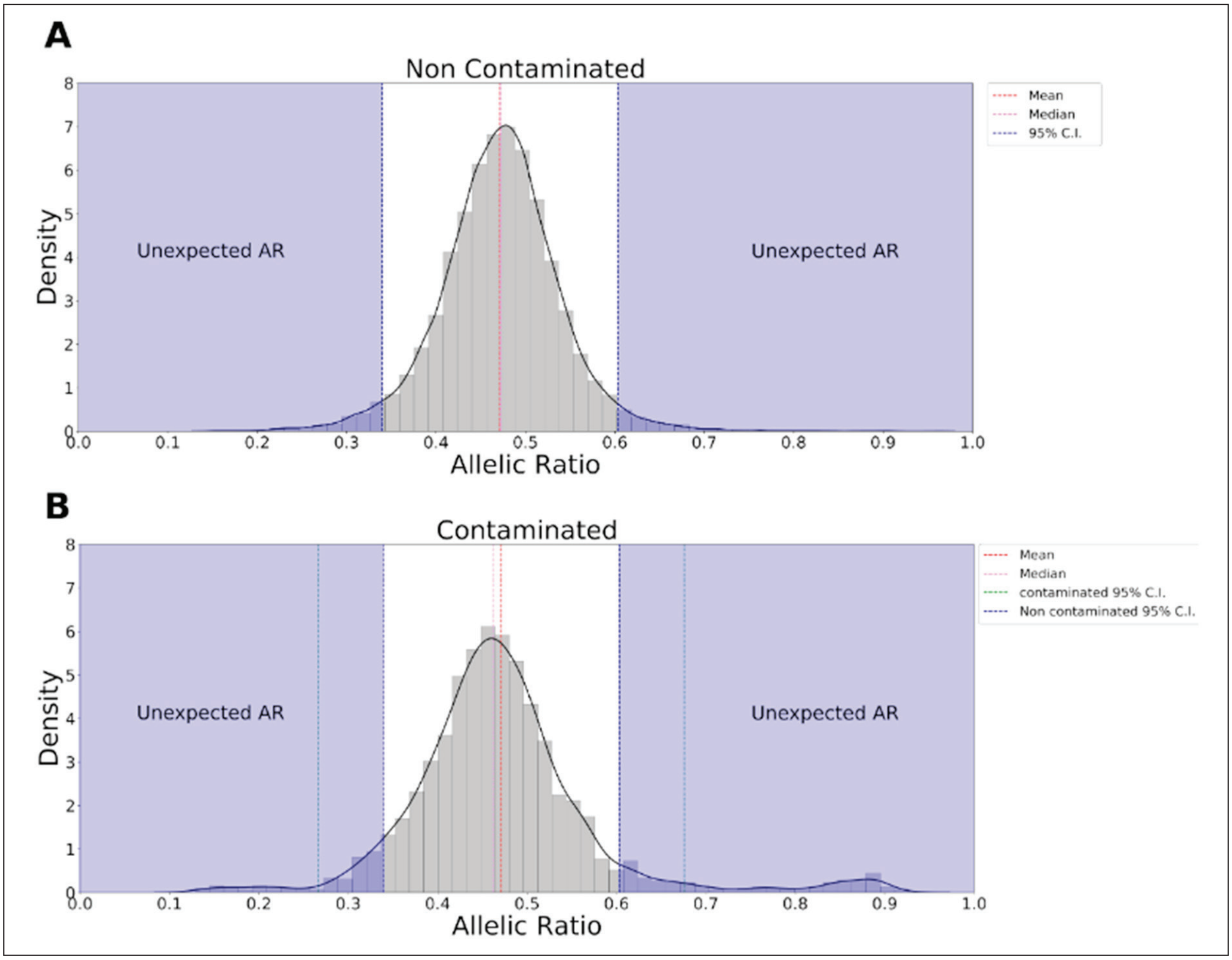

Figure 1. Distribution of allele ratios AR in reference dataset and contaminated samples. (A) Distribution of AR in reference dataset. The red line is the mean of the distribution, the violet the median and the blue lines define the 95% confidence interval (CI95). The blue areas outside the CI95 define the region of unexpected AR: whatever falls outside the CI95 is considered unexpected. It can be seen that the distribution is normal with minimal tails in the unexpected regions. (B) Distribution of AR in the artificially contaminated samples. The red line is the mean of the distribution, the violet the median, and the blue lines define the 95% confidence interval of the reference dataset, the green dotted line defines the CI95 of the contaminated dataset. The distribution is no longer normal and a greater percentage of data falls in the unexpected regions with respect to the reference dataset, showing the effects of contamination on the AR of SNPs

easy to detect contamination as low as 20% without the risk of calling a false positive, and suggests a limit of detection around 10%, a threshold at which there may be more FPs, since the upper limit of the CI95 of the reference dataset is between the score obtained by samples with 7% and 10% contamination. The z-score % threshold for calling a contaminated sample should therefore be set according to the number of FPs that can be tolerated. In order to estimate FPs at the different contamination levels, more experiments are needed. Fig. 3 shows the distribution of AR of the samples used to generate contaminated samples and the distribution of AR of the contaminated samples for different contamination percentages. The line of best fit shows that as the percentage of contamination decreases, it becomes more and more difficult to distinguish contaminated and clean samples, since the change in the distribution of AR is minimal for low

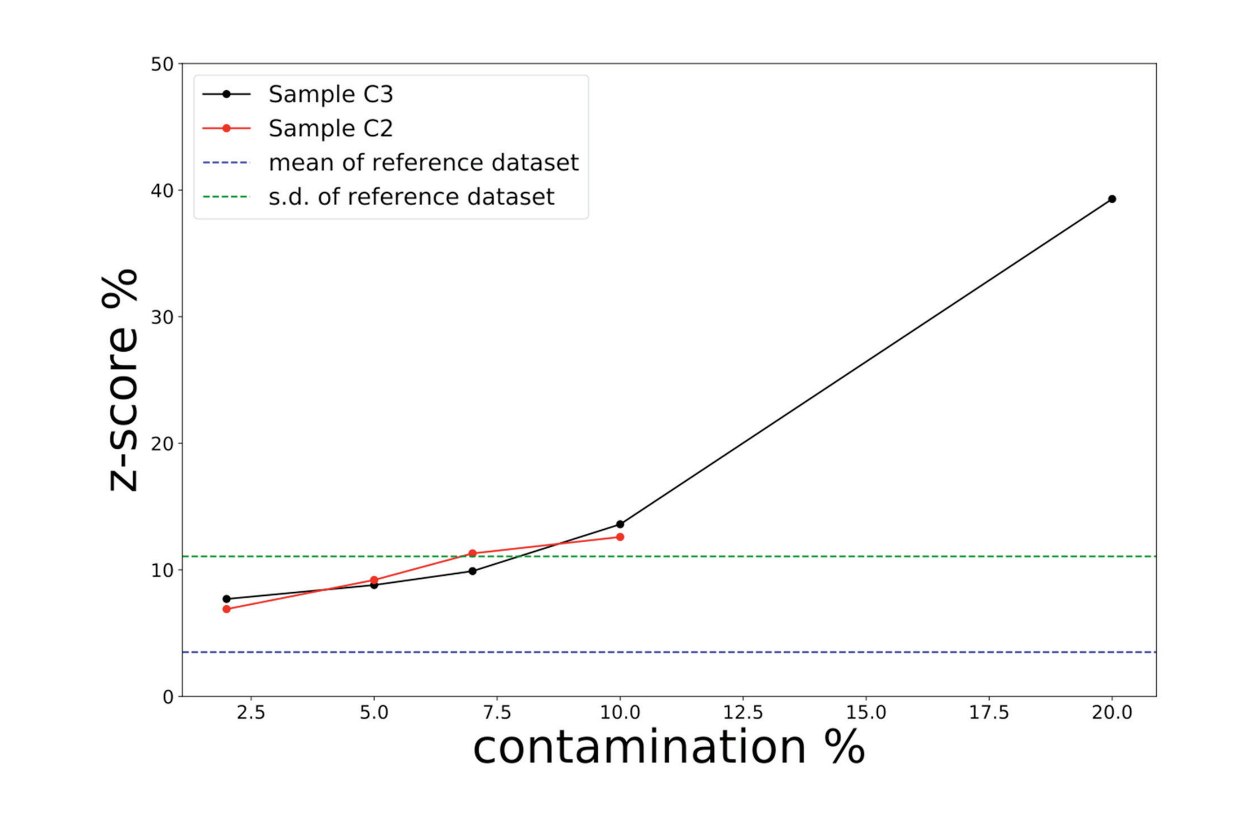

Figure 2. Percentage of SNPs with z-score outside the defined thresholds (-1.96/+1.96) for samples of the validation set. In black the score obtained by sample C3 (see Table 1) at different ratios of contamination. In red the score obtained by sample C2 (see Table 1) at different ratios of contamination. The blue line is the mean of the z-scores obtained by the reference dataset while the green dotted line defines the upper limit of the CI95 of the z-scores of the reference dataset. The graph suggests that our method is able to detect contamination down to 20-10%. The threshold for discriminating between contaminated/non contaminated sample should be chosen depending on how many FPs can be tolerated. Contamination around 20% would probably generate no FPs. Detection of lower contaminations is possible but with more FP calls. More experiments are needed to have an estimate of FPs for different contamination percentages

contamination levels. At 5% contamination, the line of best fit of contaminated samples almost matches that of non-contaminated samples, making it impossible to differentiate the two distributions.

Discussion

Being able to detect contamination in DNA samples is of primary importance in a diagnostic environment. The method described in this paper is designed to determine from the VCF file of a sample, whether the sample is contaminated. The results show that the method is capable of clearly distinguishing contamination as low as 20% with high accuracy. Lower contaminations, down to 10%, can be detected as well, but with a higher FP rate, since part of the reference dataset showed a score similar to samples with 7-10% contamination. Contamination below 7% cannot be efficiently detected with this method, since at these levels, the distribution of AR almost exactly matched that of the reference dataset. However, for such low contamination the impact on the sample is minimal (Fig. 3A), making the probability of genotype misclassification and false positives very low.

Our method has various features that make it ideal for use as a quality control tool in diagnostic environments. It is very easy to implement, requiring only ba-

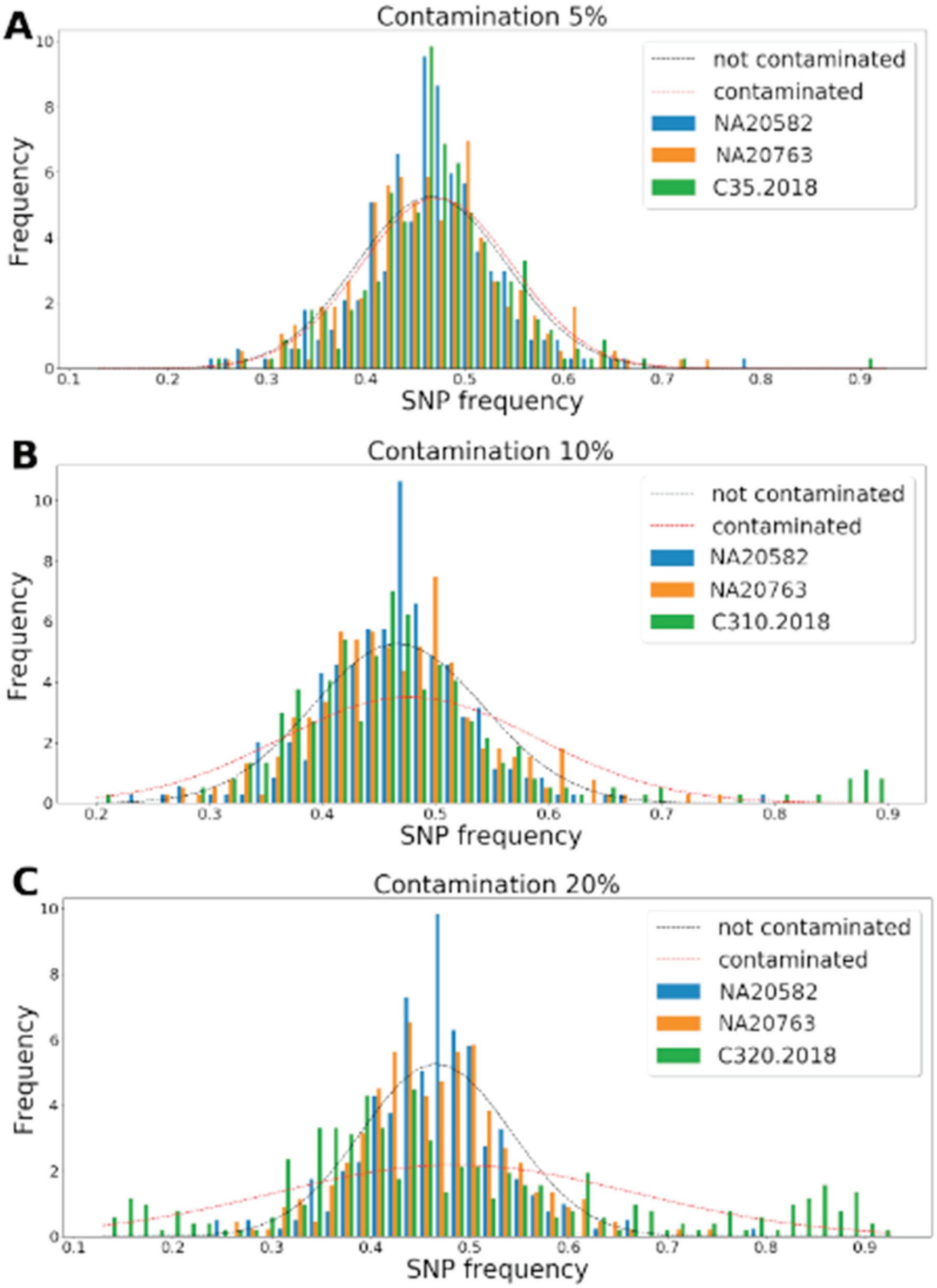

Figure 3. Comparison of AR distributions of Coriell samples used to generate contaminated samples and the resulting contaminated samples. In blue the sample used as principal, in orange the one used as contaminant and in green the resulting contaminated sample. Black dotted line is the line of best fit of the reference sample and the red line is that of the contaminated sample. (A) Results for 5% contamination. The best fit lines show that is difficult to distinguish the AR distributions, making it impossible for our method to detect contamination at such a low percentage. (B) Results for 10% contamination. The algorithm is able to distinguish the two distributions but since the score obtained by the non-contaminated sample is close to that of the 10% contaminated sample, we cannot exclude the presence of FPs if the threshold is chosen to detect 10% contamination. (C) Results for 20% contamination. In this case the contaminated sample has almost tri-modal distribution which makes it extremely easy to distinguish from the reference distribution

sic knowledge of statistics and information technology for correct implementation. It is also a fast algorithm that can analyze hundreds of samples in minutes, making it ideal for analysis of big datasets. Finally, since only the VCF files of samples are used as input, the method can easily be implemented in a NGS pipeline with minimum impact on execution time and resource consumption. It is an ideal tool for improving quality control of NGS data and the robustness of clinical results.

Conflict of interest: Each author declares that he or she has no commercial associations (e.g. consultancies, stock ownership, equity interest, patent/licensing arrangement etc.) that might pose a conflict of interest in connection with the submitted article

References

1. Sanger F, Coulson AR. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. J Mol Biol 1975; 94: 441.
2. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. PNAS 1977; 74: 5463–7.
3. Komlosi K, Solyom A, Beck M. The role of next-generation sequencing in the diagnosis of lysosomal storage disorders. J Inborn Errors Metab. Screen. 2016; 4: 2326-4594.
4. Jamuar SS, Tan EC. Clinical application of next-generation sequencing for Mendelian diseases. Hum Genomics 2015; 6: 9-10.
5. Buermans HPJ, den Dunnen JT. Next generation sequencing technology: Advances and applications. Biochim. Biophys. Acta 2014; 10: 1932–41.
6. Jun G, Flickinger M, Hetrick KN, et al. Detecting and estimating contamination of human DNA samples in sequencing and array-based genotype data. Am J Hum Genet 2012; 91: 839-48.
7. Scherczinger CA, Ladd C, Bourke MT, et al. A systematic analysis of pcr contamination. J Forensic Sci 1999; 44: 1042–5.
8. Pickrahn I, Kreindl G, Müller E, et al. Contamination incidents in the pre-analytical phase of forensic DNA analysis in Austria—Statistics of 17 years. Forensic Sci Int Genet 2017; 31: 12-8.
9. Patel RK, Mukesh J. NGS qc toolkit: A toolkit for quality control of next generation sequencing data. PLOS One 2012; 7: 1–7.
10. Lee I, Chalita M, Ha S, Na S, Yoon S, Chun J. Contest16s: an algorithm that identifies contaminated prokaryotic genomes using 16S RNA gene sequences. Int J Syst Evol Microbiol 2017; 67: 2053–7.
11. Marceddu G, Dallavilla T, Guerri G, Manara E, Chiurazzi P, Bertelli M. Pipemagi: an integrated and validated workflow for analysis of NGS data for clinical diagnostics. Eur Rev Med Pharmacol Sci 2019; 23: 6753–65.
12. Kluyver T, Ragan-Kelley B, Pérez F, et al. Jupyter notebooks – A publishing format for reproducible computational workflows. 20th International Conference on Electronic Publishing, 2016.
13. McKinney W. Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference 2010; 445: 51–6.
14. McKinney W. Pandas: a foundational python library for data analysis and statistics. Python High Performance Science Computer, 2011.
15. Seabold S, Perktold J. Statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference, 2010.
16. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng 2007; 9: 90–5.

Received: 3 September 2020
Accepted: 14 October 2020

Correspondence:
Giuseppe Marceddu
Via Maso della Pieve, 60/A
Rovereto (TN), Italy
E-mail: giuseppe.marceddu@assomagi.org