Flagging incorrect nucleotide sequence reagents in biomedical papers: To what extent does the leading publication format impede automatic error detection?

Cyril Labbé, Guillaume Cabanac, Rachael A West, Thierry Gautier, Bertrand Favier, Jennifer Byrne

Scientometrics (2020) 124:1139–1156
https://doi.org/10.1007/s11192-020-03463-z

Flagging incorrect nucleotide sequence reagents in biomedical papers: To what extent does the leading publication format impede automatic error detection?

Cyril Labbé[1] · Guillaume Cabanac[2] · Rachael A. West[3,4] · Thierry Gautier[5] · Bertrand Favier[6] · Jennifer A. Byrne[7,8]

Received: 1 October 2019 / Published online: 22 May 2020

Abstract
In an idealised vision of science the scientific literature is error-free. Errors reported during peer review are supposed to be corrected prior to publication, as further research establishes new knowledge based on the body of literature. It happens, however, that errors pass through peer review, and a minority of cases errata and retractions follow. Automated screening software can be applied to detect errors in manuscripts and publications. The contribution of this paper is twofold. First, we designed the erroneous reagent checking (ERC) benchmark to assess the accuracy of fact-checkers screening biomedical publications for dubious mentions of nucleotide sequence reagents. It comes with a test collection comprised of 1679 nucleotide sequence reagents that were curated by biomedical experts. Second, we benchmarked our own screening software called Seek&Blastn with three input formats to assess the extent of performance loss when operating on various publication formats. Our findings stress the superiority of markup formats (a 79% detection rate on XML and HTML) over the prominent PDF format (a 69% detection rate at most) regarding an error flagging task. This is the first published baseline on error detection involving reagents reported in biomedical scientific publications. The ERC benchmark is designed to facilitate the development and validation of software bricks to enhance the reliability of the peer review process.

Keywords Scientific text · Biomedical literature · Fact-checking · Errors · Nucleotide sequences · Reagents · Genes · Benchmark · PDF

Introduction

The original purpose of a scientific paper is to be read. Reading scientific papers is the main way that scientists acquire knowledge (Volentine and Tenopir 2013). Most publishing houses format and deliver papers in Portable Document Format a.k.a. PDF. Each PDF results from a workflow connecting authors with readers. This starts with authors providing their paper:

✉ Cyril Labbé
cyril.labbe@univ-grenoble-alpes.fr

Extended author information available on the last page of the article

usually multiple Microsoft Word or LaTeX files (Brischoux and Legagneux 2009). Submissions must often follow stringent presentation rules defined in authors' guidelines which refer to font size, reference formatting, and other parameters and specifications (Hartley and Cabanac 2017). Publishers ingest this raw material (Morris et al. 2013, pp. 116–117) and transform this into a standard format called XML-JATS, the *Journal Article Tag Suite* (Beck 2011), which is then transformed on-demand into multiple presentation formats: browsers receive HTML and readers can download and print in PDF to get the look-and-feel of journal issues off the press.

In addition to *human reading*, literature-based knowledge discovery involves performing *distant reading* based on text- and data-mining algorithms implemented in various software (Bruza and Weeber 2008). Automation is necessary as 1.3 million papers are published worldwide each year (Soete et al. 2015, p. 36). Text mining also enhances information retrieval tasks by automatically identifying and indexing named entities, such as genes (Galea et al. 2018) and species (Gerner et al. 2010).

For both humans and machines, it is highly critical that the knowledge conveyed in scientific papers is sound and valid as per the current body of knowledge. Nonetheless, part of the literature is plagued with various errors and questionable statements, as detected by fact-checking endeavours and error sleuths (Ledford et al. 2017, p. 321). Although individuals manage to *visually* screen biomedical images from thousands of papers (Baker 2016; Christopher 2018; Van Noorden 2015), the enormous worldwide scientific production calls for automated approaches relying on machine learning, as described by Acuna et al. (2018), for instance. These automated approaches include the detection of plagiarism (Citron and Ginsparg 2015), *p*-value hacking (Nuijten et al. 2016), and discrepancies between the claims presented and current knowledge on the topic (Labbé et al. 2019).

For both knowledge discovery and fact-checking, it is to be expected that the quality and format of the input texts will affect the results produced by automatic text screening approaches. The aim of this paper is to measure the extent to which various input formats can affect the quality of error detection. Based on the benchmark that we develop in this paper, our main finding is that the leading file format for publications, which is PDF, is not the most appropriate format to perform both knowledge discovery and fact-checking. This calls for the release of scientific papers in machine-readable formats such as XML-JATS.

This paper is organised as follows. The first section introduces the research issue that we tackle: the need to flag errors in the biomedical literature. We hypothesise that the format of input materials will produce results of varying quality. To measure the extent to which the input format impedes error detection, we designed the original benchmark called `ERC` (erroneous reagent checking) presented in the second section. This is a generic benchmark released as supplementary material (see "Appendix `ERC` benchmark `ERC_H_v2` test collection") allowing the assessment of *any* fact-checker, namely systems that aim to spot errors in biomedical papers. We ran this benchmark on our fact-checking system called Seek&Blastn (Labbé et al. 2019). The third section reveals that automatic error detection from materials formatted in PDF—the leading publishing format—is sub-optimal. This leads us to stress the need to perform fact-checking on text-preserving formats such as XML-JATS which is already employed by scientific publishers.

Research issue: flagging errors in papers

With the increase in scientific publications and retractions (Brainard and You 2018), there is an urgent need to automate error screening to improve the quality of the literature. The people who engage in this text-mining task generally use materials available online: the articles in PDF, as the *de facto* standard. Most error-detection software, however, rely on plain text analysis. As a result, systems need to disentangle the textual content from the presentation markers such as layout, logos, headers, footers, and so on. Many programs strive to extract plain text from PDFs: Bast and Korzen (2017) tabulated 14 programs such as the well-known pdftotext,[1] Grobid (Lopez 2009), and Icecite (Bast and Korzen 2013). Bast and Korzen (2017) stress how challenging this task remains today.

Given that text extraction from a PDF file often proves problematic (Bast and Korzen 2017), we as text-miners legitimately wonder about the extent to which a degraded text lowers the quality of results for a particular error detection tool. In this paper we tackle this research question to quantify the impact of degraded texts on the performance of an error detection tool.

The `ERC` benchmark that we designed assesses error detection failures attributed to degraded input. As a case study, we ran this benchmark on Seek&Blastn, thus extending our published research on flagging errors in the biomedical literature (Byrne and Labbé 2017). In this field of science, authors report their experiments with details about the biological materials employed (e.g., nucleotide sequences, cell lines), stressing their key properties. Expert readers are able to check the claims for errors such as where a biological material is said to have particular properties which are not supported by state-of-the-art resources, such as knowledge bases (Labbé et al. 2019).

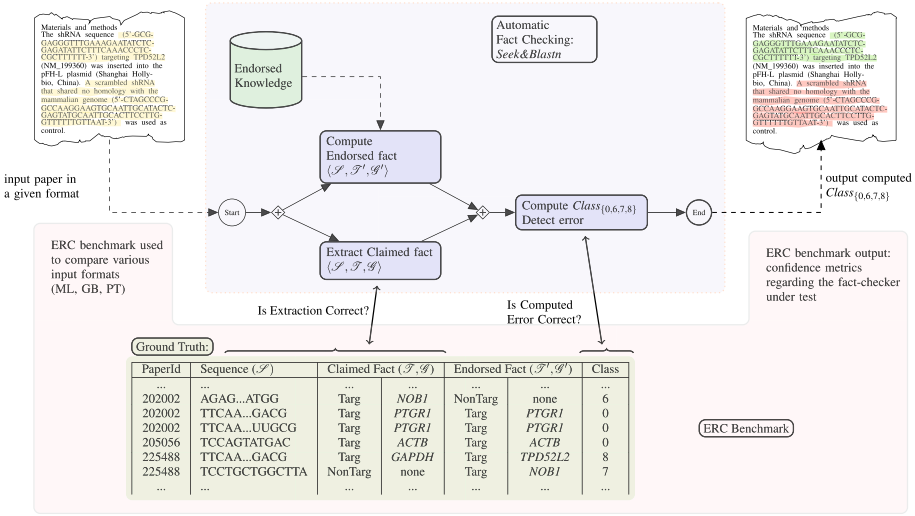
Since it is likely that the performance of error detection software depends on the quality of the text input, one needs to perform failure analysis in order to attribute poor performance to (1) the poor quality of input text and/or (2) flaws related to the fact-checking algorithm. This is the rationale behind the twofold contribution of this paper. First, the next section specifies the `ERC` benchmark. Second, we run this benchmark on the `ERC_H_v2` biomedical test collection whose corpus comes in three formats: (i) native markup language (XML-JATS or HTML), (ii) inferred markup language from PDFs, and iii) extracted text from PDFs. We then comment on the varying quality of error detection with regards to the input text format used.

Contribution 1: Benchmarking the error flagging task

There is a long tradition of evaluation in Information Retrieval, namely the field dealing with the design of search engines. So-called 'test collections' were designed to enable the automatic, systematic, reproducible evaluation of search engines (Voorhees 2007). A test collection is comprised of:

- Input of the system under study: a textual corpus.
- Ground truth: the expected output of the system under study.
- Metrics to assess the quality of a given output regarding the expected output.

[1] http://www.xpdfreader.com/pdftotext-man.html.



PaperId	Sequence ($\mathscr{S}$)	Claimed Fact ($\mathscr{T}$,$\mathscr{G}$)		Endorsed Fact ($\mathscr{T}'$,$\mathscr{G}'$)		Class
...	...	...	...	...	...	...
202002	AGAG...ATGG	Targ	*NOB1*	NonTarg	none	6
202002	TTCAA...GACG	Targ	*PTGR1*	Targ	*PTGR1*	0
202002	TTCAA...UUGCG	Targ	*PTGR1*	Targ	*PTGR1*	0
205056	TCCAGTATGAC	Targ	*ACTB*	Targ	*ACTB*	0
225488	TTCAA...GACG	Targ	*GAPDH*	Targ	*TPD52L2*	8
225488	TCCTGCTGGCTTA	NonTarg	none	Targ	*NOB1*	7
...	...	...	...	...	...	...

Fig. 1 Synoptic view of the ERC benchmark introduced in this paper, designed to assess the quality of automatic fact-checking in the biomedical literature. A scientific paper is input to the error-detection software. *Extracted Claimed facts* are extracted from the paper: each one is a nucleotide sequence with its claimed properties (non-targeting vs. targeting with associated target). It is then matched to *Computed Endorsed facts* stemming from accessing an endorsed knowledge source. This comparison leads to a *Computed* $Class_{\{0,6,7,8\}}$ for each nucleotide sequence. The benchmark aims to compute quality metrics by comparing system outputs to the *Ground Truth*

This framework enables one to:

1. Benchmark the quality of a system compared with other systems, with a fixed input.
2. Check the performance gain/loss of a new system configuration compared to a baseline configuration, with a fixed input.
3. Benchmark the quality of a fixed system fed with varying inputs.

The ERC error-detection benchmark that we introduce in this section addresses these three purposes. In the subsequent experiments section, we report results according to point (3) above as we assess the quality of Seek&Blastn regarding varying inputs.

To the best of our knowledge, there is no existing benchmark dealing with the error detection in the biomedical literature. Our original benchmark depicted in Fig. 1 aims to address this issue. Let us introduce a handful of biomedical concepts prior to explaining this figure—they appear in *italics* in the following. We consider scientific papers in the field of life sciences reporting *gene*-related experiments. These papers mention *reagents*: sequences of *nucleotides*, each nucleotide being represented as a letter among A, T, C, G, and U. Each reagent may or may not bind to a target within the genome or transcriptome. The presence or absence of binding depends on the *homology* between the reagent and its target, which is typically a defined place/localisation in the genome, such as a named gene. A gene is identified with a standard name (e.g., *NOB1* and *TPD52L2*)

and the reagent–gene homology can be assessed using the BLASTN software (Altschul et al. 1997). A reagent is said to be:

- *targeting* gene X when it exhibits *a significant level of homology or identity* with gene X.
- *non-targeting* when it exhibits *no significant homology* with any gene or sequence of the stated genome.

The rule-based algorithm illustrated in Fig. 2 of the "Appendix" (Reagent–gene assessment of nucleotide sequence homology) assesses the significance of homology, leading to predictions of whether reagents are likely to be targeting or non-targeting. These rules have been determined and validated on the base of laboratory practices (BF, JAB, and TG) and on common siRNA and shRNA rule design (Yu et al. 2002). Deciding on the (non-)targeting nature of a reagent requires feeding this rule-based algorithm with the output of the BLASTN software (run with the reagent given as input). An *endorsed fact* (Computed Endorsed fact in Fig. 1) reflects the current knowledge provided from BLASTN and refined by the rule-based algorithm.

In biomedical publications, nucleotide sequence reagents are claimed to be (non-) targeting a gene. For various reasons a claim may be wrong (Labbé et al. 2019), such as through typographical errors, copy and paste errors or through limited understanding of the experiments described. Invalidating a claim requires a comparison of an automatically extracted *claimed fact* with the *BLASTN endorsed fact*. This fact-checking process tags each sequence with one of the four[2] following classes:

- Class_0: *supported claim*. The nucleotide sequence in the text and its associated claim is valid according to current knowledge.
- Class_6: *unsupported claim of targeting status*. A nucleotide sequence is said to be targeting but is predicted to be non-targeting according to current knowledge.
- Class_7: *unsupported claim of non-targeting status*. A sequence is said to be non-targeting but is predicted to be targeting according to current knowledge.
- Class_8: *targeting claim supported but incorrect target*. A stated targeting nucleotide sequence is predicted to target a different gene or nucleotide sequence to that claimed.

Papers with unspecific targets or claimed status were removed from the ground truth.

The `ERC` benchmark aims to measure the quality of a fact-checking system by comparing its output to a test collection. The test collection stores nucleotide sequences that human experts tagged with a $\text{Class}_{\{0,6,7,8\}}$. Benchmarking a fact-checking system consists of comparing its output to the expected answer for each sequence. The metrics we defined in `ERC` are run on the output of a fact-checking system: one of these classes (0, 6, 7, or 8) or a 'no-decision' answer for each sequence. This latter case occurs when the system is not able to provide an answer and reports this to end-users.

Metrics are defined to assess the overall performance of the system relying on three chained processes (CP1 to CP3) that one needs to evaluate separately. Each metric combines quantities from the set of variables introduced in Table 1. For each CP, we distinguish three metrics: either the system is successful (metric OK) or the system either fails to

[2] Other classes were also considered but not used in this paper. The description of $\text{Class}_{\{1,2,3,4,5,9\}}$ appears in the supplementary materials ("Appendix `ERC` benchmark `ERC_H_v2` test collection").

Table 1 Variables used to define the metrics of the ERC fact-checking benchmark

Symbol	Definition
s	# of nucleotide sequences present in the corpus
c	# of nucleotide sequences correctly extracted from the corpus
f	# of extracted nucleotide sequences that are alien to the corpus. For example, a text mining tool might extract a reagent appearing across two columns as two different reagents, which is incorrect
a	# of correctly extracted nucleotide sequences whose status was correctly assigned by the fact-checking tool
n	# of nucleotide sequences for which the system failed to assign any status
w	# of nucleotide sequences for which the system assigned the wrong status
a'	# of correctly extracted nucleotide sequences whose target was correctly assigned
n'	# of nucleotide sequences for which the system failed to assign any target
w'	# of nucleotide sequences for which the system assigned the wrong target
o	# of nucleotide sequences for which the system output is correct: either Fact-checked as Class_0 or error flagged as $\text{Class}_{\{6,7,8\}}$
p	# of nucleotide sequences for which the system failed to take a decision
q	# of nucleotide sequences for which the system assigned a wrong decision (i.e., incorrect class assigned)

extract the information to be checked (metric KO1), or it fails to check correct information (metric KO2, wrong decision is made). These metrics in the [0, 1] range are computed as follows:

CP1. *Sequence*:

- Sequence_OK = c/s is recall-oriented and reflects the ability to extract all the nucleotide sequences from the corpus.
- Sequence_KO1 is not computed as the nucleotide sequence is either correctly extracted (i.e., Sequence_OK) or missed (i.e., Sequence_KO2).
- Sequence_KO2 = $f/(f+c)$ is precision-oriented and reflects the trust that the user can place in the results of the sequence extraction task.

CP2. *Status* for each correctly extracted nucleotide sequence:

- Status_OK = a/c reflects the ability to automatically assign the claimed status to a nucleotide sequence reagent: non-targeting vs targeting.
- Status_KO1 = n/c measures the proportion of nucleotide sequences for which the fact-checking tool failed to assign a specific status.
- Status_KO2 = w/c measures the proportion of nucleotide sequences for which the fact-checking tool misassigned the claimed status.

CP3. Targeted *gene or sequence* for each targeting nucleotide sequence:

- Gene_OK = a'/c measures the proportion of correctly extracted sequences to which the claimed gene identifier was associated (*none* for non-targeting sequences).
- Gene_KO1 = n'/c measures the proportion of correctly extracted nucleotide sequences to which no gene identifier was associated (value *unknown*).
- Gene_KO2 = w'/c measures the proportion of correctly extracted sequences to which a wrong gene identifier was associated (e.g., the text mentions *TPD52L2* whereas the fact-checking tool extracted a different identifier).

The fact-checking system compares (1) the text stated and extracted fact with (2) the endorsed fact to produce an output $\text{Class}_{\{0,6,7,8\}}$ for a given nucleotide sequence. An error while performing CP1, CP2, or CP3 is responsible for a false output from the fact-checking system. We measure the end-to-end performance of the benchmarked tool as:

- Fact-check_OK = o/s measures the proportion of correctly checked nucleotide sequences.
- Fact-check_KO1 = p/s measures the proportion of nucleotide sequences for which the fact-checking process did not produce a decision.
- Fact-check_KO2 = q/s measures the proportion of nucleotide sequences for which a wrong decision was made (e.g. Class_0 instead of Class_8).

At this point, the metrics-defined can be used to answer the main question of this paper, namely what is the performance decay (if any) when providing inputs in PDF format compared to other, more structured, formats? We answer this question by comparing Fact-check_OK across all tested input formats. Running through the whole processing chain (CP1, CP2, and CP3) indicates where performance decreases. This helps *system designers* to decide where to focus their future efforts.

People who employ the fact-checking system include biologists, journal staff, and text miners. Detecting errors in the papers that they read is crucial to not trust erroneous literature. From the end-users' perspective, flagging errors automatically can prove risky, as no system is perfect. Reporting a trust level for each output of the detector (i.e., $\text{Class}_{\{0,6,7,8\}}$) can help end-users to assess the trustworthiness of the system result. This is why the benchmark provides the following extra metrics regarding the success rate per error class.

As a reminder, each nucleotide sequence in the ground truth is tagged with one expected output ($\text{Class}_{\{0,6,7,8\}}$). Due to this partition, the number of sequences s is the sum of the number of sequences in each class, that is $s = \sum_{i \in \{0,6,7,8\}} s_i$ where s_i is the number of sequences for Class_i in the test collection. The following numbers are useful to measure the accuracy of a system under test:

- o_i is the number of nucleotide sequences of Class_i that were correctly reported by the system.
- u_i is the number of nucleotide sequences of Class_i for which no decision was taken by the system.
- m_i is the number of nucleotide sequences of Class_i that were incorrectly reported by the system.

For each Class_i, we define the following metrics:

- Class_i_OK = o_i/s_i is the proportion of nucleotide sequences for which the system output matches the expected class given in the Ground Truth.
- Class_i_KO1 = u_i/s_i is the proportion of nucleotide sequences for which the system was unable to associate a class ('no-decision' was reported).
- Class_i_KO2 = m_i/s_i is the proportion of nucleotide sequences for which the system produced a wrong class. The distribution of these wrong decisions among the different classes can then be computed to identify the most frequent erroneous pairs of Class_i and Output_i.

$Class_i$_KO2 is crucial to highlight situations when the fact-checking software confused one class for another one. For example, the $Class_0$ sequences (endorsed facts) may be confused with $Class_8$ sequences (gene mismatch). This measure reflects the likelihood of misclassifying a nucleotide sequence of a given class. This informs the end-user about the level of confidence (s)he may have with regards to each type of output.

The next section reports the results of the benchmark that we performed on the `ERC_H_v2` test collection that we built and distributed for reproducibility concerns.

Contribution 2: Benchmarking various text input formats

Deriving the `ERC_H_v2` test collection

The benchmarking of fact-checking software requires the assembly of a test collection that represents a variety of errors. This is challenging as most of the literature is supposed to be free from such errors! We therefore designed a threefold strategy to identify error-prone papers:

1. Two of the authors (JAB and CL) reported 48 highly similar publications that commonly described nucleotide sequence reagents (Byrne and Labbé 2017). JAB fact-checked a proportion of the nucleotide sequence reagents described in these papers using BLASTN.
2. The list of aforementioned dubious sequences was used as query to Google Scholar to retrieve additional, potentially questionable, papers. We hypothesised that some erroneous papers may stem from paper mills (Liu and Chen 2018) that are known to produce papers with similar textual contents. Hence, we used the PubMed *Similar articles* feature[3] to retrieve the most similar papers to a given one according to their textual contents. This resulted in the corpus of 155 papers used in (Labbé et al. 2019).
3. All 48 papers flagged in Step 1 were used as seeds (i.e., starting points) to collect 1,664 similar papers using the PubMed *Similar articles* feature.

We manually screened these error-prone papers to constitute the ground truth illustrated in Fig. 1.

In a preliminary analysis, four experts (BF, JAB, TG, and Ms Natalie Grima (Labbé et al. 2019)) annotated 44 papers. Each paper was presented in PDF format while the markup-language counterpart was also downloaded in XML-JATS or HTML for further processing. Experts performed manual fact-checking, involving paper reading to identify nucleotide sequences, delineation of the purpose of each nucleotide sequence in the reported experiments, as well as the analysis of the BLASTN results they obtained. Among the annotated 44 papers mentioning 77 error-prone sequences, 21 sequences from 12 papers triggered discussions and all conflicting cases were resolved after reaching consensus. This preliminary analysis stressed two kinds of issues. First, the wording of some

[3] http://ncbi.nlm.nih.gov/books/NBK3827/#pubmedhelp.Computation_of_Similar_Articl.

papers was either ambiguous or difficult to understand. Second, all papers mentioning any nucleotide sequence not related to the human genome or transcriptome required the identification of the appropriate nucleotide sequence database for reagent fact-checking.

Following this preliminary analysis, we decided to focus on papers where all nucleotide sequence reagents relate to the human genome or transcriptome.

The ground truth capturing expert knowledge was acquired as follows. Biomedical researchers (JAB, RAW, and Natalie Grima) annotated a subset of 161 papers among the previously collected ones. The annotators' assessment of each nucleotide sequence was recorded in the ground truth pictured in Fig. 1, namely the claimed characteristic targeting or non-targeting with gene name or target where applicable, the endorsed fact, and the output $\text{Class}_{\{0,6,7,8\}}$. Eventually, the ground truth resulted from the analysis of 161 papers containing 1679 nucleotide sequences annotated by at least one expert (JAB, RAW, or Natalie Grima).

All papers were processed by Grobid and pdftotext to extract two plain text versions. This resulted in three different input formats denoted as:

1. ML: markup-language (either XML or HTML).
2. GB: plain text extracted using Grobid v. 0.5.2 released October 17, 2018.
3. PT: plain text extracted using pdftotext v. 0.78.0 released June 26, 2019.

These constitute the test collection of the proposed benchmark, called `ERC_H_v2` for Error Reagent Collection, Human, version 2. We used `ERC_H_v2` to report the results of benchmarking the Seek&Blastn tool in the next section. Note that these results do not generalise to the entire biomedical literature, as the ground truth is biased towards error-prone papers.

Assessing the efficiency of the fact-checking Seek&Blastn tool

Running the `ERC` benchmark allowed us to examine the extent to which input format impedes the flagging of errors in the biomedical literature. All metrics reported in Table 2 were computed for the three different input formats: ML, GB, and PT. Overall, the share of correctly classified sequences in $\text{Class}_{\{0,6,7,8\}}$ reflects the performance of the fact-checking on a given input format (see the aforementioned Fact-check_OK definition). ML is the most appropriate format for Seek&Blastn with a 79% success rate (Fact-check_OK) when processing text in HTML or XML formats. Grobid (GB) appears to be a viable alternative with a 69% success rate (Fact-check_OK) on the text extracted from PDFs with this software. Fact-checking effectiveness decreases to 58% of all sequences correctly classified based on the text produced with pdftotext (PT).

Let us unpack the overall performance according to the three underlying chained processes CP1, CP2, and CP3 :

- Sequence detection (CP1) performs the best on ML input with 96% Sequence_OK compared to GB and PT characterised by a lower Sequence_OK of 87% and 79% respectively. As a reminder, each ATCGU sequence is made up of 9 to 100+ characters; some authors use spaces to group nucleotides in blocks. Nucleotide sequences appear as described above in ML but may be altered when formatted in PDF. Sequences may be split across columns or pages, which can then mix them up with the headers and footers of the pages.

- With the correctly identified sequences as input:
  - Status identification (CP2) is also more accurate for ML and GB (93% and 92% Status_OK) compared to PT (88% Status_OK). CP2 is a difficult task, requiring the parsing of the sentence or table featuring each sequence. The variability and structural ambiguity of either natural language or table layouts makes CP2 particularly challenging. The tied performance of ML and GB suggests that text extraction with GB managed to preserve text structure. In contrast, PT produced a less suitable text for Seek&Blastn to identify sequence status correctly.
  - Gene identification (CP3) with ML or GB inputs (66% and 64% Gene_OK) outperforms PT (49% Gene_OK). This task first identifies gene names out of the tokenized sentence, where there can be variability in naming genes and spelling gene identifiers (Giorgi and Bader 2020). Then the associated nucleotide sequence must be identified. Results reflect that CP3 is the most complex task. Seek&Blastn clearly needs to be improved in this direction. Again, the tied performance of ML and GB suggests that text extraction with GB is a suitable alternative to ML.

Let us now focus on the 'Fact-checking failed' columns of Table 2 to perform a failure analysis. The input format has barely no influence on KO1 and KO2, which suggests that even with a well-formatted input (ML) Seek&Blastn fact-checks with the same performance as with GB or PT. For CP1, either a sequence is correctly extracted (OK) or not (KO2), hence no value provided for KO1. We split the denominators in KO2 into two parts reflecting (1) the number of altered (e.g., truncated) nucleotide sequences and (2) the number of unaltered nucleotide sequences. This number (i.e., unaltered nucleotide sequences) becomes the denominator of the metrics used for CP2 and CP3 to identify the source of errors. CP3 is the most poorly performed task, which suggests that more engineering should be devoted to enhance the detection of gene identifiers within text. A noticeable figure of 58% (Fact-checking succeeded) characterises PT: it appears that the text provided by pdftotext is not well suited for gene detection as implemented by Seek&Blastn.

Next, Table 3 reflects the confidence that the user can place in Seek&Blastn results according to the $\text{Class}_{\{0,6,7,8\}}$ associated with each sequence analysed. We tabulated the three input formats and, for each class, provided the obtained performance (OK, KO1, and KO2). For ML, Class_0 is reliable with Class_0_OK = 84%. Class_0 occurs the most frequently with $N = 1137$ out of $N = 1361$ nucleotide sequences. For KO2, the nature of the wrong decision made is also reported with its class number. This enables one to estimate which class is more likely to be confused with another one: Class_0 and Class_8 are the most challenging in this regard. The table shows that 16% of results tagged with Class_0 are failures (i.e., KO1 and KO2). When KO2 occurs, the most frequent class is Class_8 with a 79% likelihood. All inputs considered (i.e., ML, GB, PT), Class_6 appears to be the most reliable one (Class_6_OK = 86% for ML) and Class_8 is the least reliable one (Class_8_OK = 65% for ML).

Users of the benchmarked fact-checking software are expected to benefit from the reports (Tables 2, 3) highlighting the strengths and weaknesses expected. In addition, these reports enable the designers of fact-checking software to assess the accuracy of their software; they also reveal cases of low performance that should be further investigated. The contribution of this paper is twofold: defining the `ERC` benchmark and populating the expert-annotated `ERC_H_v2` test collection. We expect this latter component to help fact-checking designers to create and tune their systems to reach higher accuracy. In closing, let us stress the sensitivity of most classification approaches to class imbalance which is a well-known problem in machine learning (Japkowicz and Stephen 2002). This

Table 2 Effectiveness of fact-checking broken down at sub-task levels (Sequence, Status, and Gene) for 1679 nucleotide sequences occurring in papers considered in three formats: Markup language provided by publishers (ML) versus text extracted by running Grobid (GB) and pdftotext (PT) on PDF files

(Sub)-task	Fact-checking succeeded			Fact-checking failed					
	OK: Correct detection *right decision made*			KO1: Fact not checked *no decision made*			KO2: Incorrect result [a] *wrong decision made*		
	ML	GB	PT	ML	GB	PT	ML	GB	PT
Overall Fact-check	.79 1322/1679	.69 1160/1679	.58 974/1679	.04 68/1679	.05 78/1679	.07 116/1679	.13 219/1679	.13 220/1679	.14 229/1679
CP1 Sequence	.96 1609/1679	.87 1458/1679	.79 1319/1679	—	—	—	.05 88/(88+1609)	.09 151/(151+1458)	.22 363/(363+1319)
CP2 Status	.93 1502/1609	.92 1338/1458	.88 1167/1319	.05 76/1609	.06 89/1458	.09 124/1319	.02 31/1609	.02 31/1458	.02 28/1319
CP3 Gene	.66 1068/1609	.64 938/1458	.49 640/1319	.22 346/1609	.22 315/1458	.35 462/1319	.12 195/1609	.14 205/1458	.16 217/1319

[a] As defined earlier, KO2 denominators are split into two parts: 1) the number of altered (e.g., truncated) nucleotide sequences and 2) the number of unaltered nucleotide sequences. This latter number becomes the denominator of the metrics used for CP2 and CP3 to identify the source of errors.

Table 3 Recall-oriented failure analysis broken down by sequence $\text{Class}_{\{0,6,7,8\}}$ for the textual passages of the papers in the `ERC_H_v2` test collection

Markup-language input (ML)

Type	Class_0 ($N = 1,361$) *no errors to be found in the input text*			Class_6 ($N = 74$) *unsupported claim of targeting seq.*			Class_7 ($N = 35$) *unsupported claim of non-targeting seq.*			Class_8 ($N = 139$) *targeting seq. claimed with wrong target*		
Detection	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2
Proportion (%)	.84	.04	.12	.86	.04	.09	.89	.06	.06	.65	.04	.32
Wrong choice was:	Class_6	Class_7	Class_8	Class_0	Class_7	Class_8	Class_0	Class_6	Class_8	Class_0	Class_6	Class_7
Proportion (%)	.15	.06	.79	.86	.00	.14	.50	.00	.50	.84	.16	.00

Grobid input (GB)

Type	Class_0 ($N = 1,236$) *no errors to be found in the input text*			Class_6 ($N = 64$) *unsupported claim of targeting seq.*			Class_7 ($N = 36$) *unsupported claim of non-targeting seq.*			Class_8 ($N = 122$) *targeting seq. claimed with wrong target*		
Detection	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2
Proportion(%)	.81	.05	.14	.86	.05	.09	.83	.11	.06	.61	.05	.34
Wrong choice was:	Class_6	Class_7	Class_8	Class_0	Class_7	Class_8	Class_0	Class_6	Class_8	Class_0	Class_6	Class_7
Proportion (%)	.12	.07	.81	.68	.00	.33	.50	.00	.50	.83	.17	.00

pdftotext (PT)

Type	Class_0 ($N = 1,124$) *no errors to be found in the input text*			Class_6 ($N = 57$) *unsupported claim of targeting seq.*			Class_7 ($N = 22$) *unsupported claim of non-targeting seq.*			Class_8 ($N = 116$) *targeting seq. claimed with wrong target*		
Detection	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2	OK	KO1	KO2
Proportion(%)	.75	.10	.16	.86	.02	.12	.82	.05	.14	.57	.06	.37
Wrong choice was:	Class_6	Class_7	Class_8	Class_0	Class_7	Class_8	Class_0	Class_6	Class_8	Class_0	Class_6	Class_7
Proportion (%)	.13	.05	.82	.86	.00	.14	.33	.00	.67	.84	.16	.00

is also characteristic of the fact-checking ground truth contributed in this paper, revealed by the sizes of each class for the text in ML format ($Class_0$: 1361, $Class_6$: 74, $Class_7$: 35, $Class_8$: 139, as seen in Table 3) While the rule-based classification implemented in Seek&Blastn (Labbé et al. 2019) makes this software resilient to data imbalance, using other techniques (e.g., Support Vector Machines) on such imbalanced classes may require advanced over/under-sampling.

Discussion

Let us now discuss the `ERC_H_v2` test collection we contributed in this paper comprising:

1. The input given to the fact-checking system under study: a textual corpus.
2. The ground truth: the expected output of the system under study.
3. Metrics to assess the quality of a given output regarding the expected output.

First, we reported the results of Seek&Blastn (Labbé et al. 2019), which is the first biomedical reagent fact-checking software to our knowledge. By releasing the benchmark with the associated `ERC_H_v2` test collection,[4] we encourage other researchers to tackle the error detection problem. Running the benchmark on the outputs of different systems will inform on the accuracy of the benchmarked approaches (e.g., machine learning vs. rule-based decision making). As such, Seek&Blastn contributes a baseline for such other systems.

Second, the results reported in Tables 2, 3 using `ERC_H_v2` concern human reagents only. Covering a larger scope of the literature would require to build a ground truth addressing multiple species (e.g., rat, mouse, zebrafish).[5] In addition, these results reflect the state-of-the-art knowledge accessed through BLASTN and the expert rules ("Appendix (Reagent–gene assessment of nucleotide sequence homology)") applied to this knowledge. The advance of science may update these in future. Such changes can be propagated to `ERC_H_v2` to reflect this evolution and the benchmark can be run again based on new versions.

Third, we conceived the performance metrics with the user and the fact-checking designer in mind. Other metrics can be computed and the chained processes can be deconstructed for fine-grained analysis. CP3 is a good candidate as it entails two subtasks: a gene entity recognition task (e.g., Galea et al. 2018, p. 2477) and an association with the reagent detected by CP1.

In summary, finding errors in the literature is a challenging task requiring the alliance of skills in text mining, natural language processing, and scientometrics. This benchmark as a whole contributes to raise awareness of this critical issue.

[4] Each paper is identified with its PMID in `ERC_H_v2` (Fig. 1). Using open access (OA) papers *only* would enhance the benchmark usability. However, this would not reflect that errors appear in both non-OA and OA papers. Due to licensing concerns, we must leave the harvesting of the plain texts to the benchmark users.

[5] This would insert another chain process—between CP2 and CP3—to extract the species name, which would be used to restrict the sequence database for the BLASTN query. While the targeting status appears close to the concerned nucleotide sequence, this is not always the case for the species under study. Furthermore both vernacular species name and Latin identification would have to be recognised by this future Seek&Blastn.

Conclusion

The introduction of bibliographic databases fostered the indexing of scientific literature (De Bellis 2009). These products have been indexing metadata only: title, abstract, keywords, and authors. Current endeavours aim to collect the entire literature in plain text. For instance, Pulla (2019) reported that the Indian JNU initiative has indexed 73 million journal articles in PDF. Specific indexing of genes and chemicals for knowledge extraction is one of the reported purposes of such a giant resource. This endeavour is in the spirit of literature-based discovery of new knowledge (Bruza and Weeber 2008).

Collecting nearly all scientific publications in a single place offers another opportunity: one could screen the entire literature for errors. Retracting dubious papers is critical for human readers as well as for automated text-mining processes: misleading reasoning and inference must be prevented to improve the integrity of science and the trust that people place in science and the scientific method. Previous work has flagged plagiarism (Citron and Ginsparg 2015), nonsensical papers (Labbé and Labbé 2013), and erroneous papers (Byrne and Labbé 2017). Thousands of papers have been retracted from the literature (Van Noorden 2014; Ledford et al. 2017).

In the present study, we have screened *published* papers. Triggering the fact-checker earlier, during the peer review process, could save time and effort by desk-rejecting the offending submission. As is currently done to combat plagiarism (Smart and Gaston 2019), fact-checkers such as Seek&Blastn could be used to screen submitted manuscripts ahead of referee assignment.

The `ERC` benchmark proposed in this paper contributes to enhance the quality of such systems. Designers can measure the failures related to each subcomponent of the fact-checker, make changes to improve the subcomponents, and then check that no regressions in performance have occurred. Running the benchmark for Seek&Blastn highlighted the superiority of plain text (ML) over text extracted from PDFs (GB and PT) for the error detection task. Hence we recommend:

- To run fact-checkers over formats that preserve text integrity (e.g., HTML, XML-JATS, LaTeX formats) instead of formats that alter text by mixing up form and content (PDF) whenever possible.
- When PDF is the only material available, use Grobid (Lopez 2009) to extract textual contents with minimum quality loss.

Literature-based knowledge discovery through text-mining benefits from open access to error-free sources. In the foreseeable future when Open Science prevails, open access to papers published in a text-preserving format is likely to be the norm and fact-checkers such as Seek&Blastn will perform at their full potential.

Acknowledgements We gratefully acknowledge the assistance of Ms Natalie Grima (Children's Cancer Research Unit), and funding from the Post-Truth Initiative, a Sydney University Research Excellence Initiative (SREI 2020) (to JAB), and from the US Office of Research Integrity Grant ORIIR180038-01-00 (to JAB and CL). This work was supported by donations to the Children's Cancer Research Unit of the Children's Hospital at Westmead.

Appendix

`ERC` benchmark `ERC_H_v2` test collection

The `ERC_H_v2` test collection of the `ERC` benchmark is available from https://doi.org/10.5281/zenodo.3773905.

Reagent–gene assessment of nucleotide sequence homology

A reagent (i.e., a nucleotide sequence defined by the letters ATGCU) is said to be targeting or non-targeting with regards to a genome or transcriptome if it fulfils some conditions. A targeting sequence has significant homology with an intended target, such that the targeting sequence is likely to bind to the target. A non-targeting sequence must have non-significant homology to the corresponding genome or transcriptome. In Seek&Blastn, the `ERC` benchmark and associated `ERC_H_v2` test collection, rules that were adopted to define significant homology are described in Fig. 2 and Table 4.

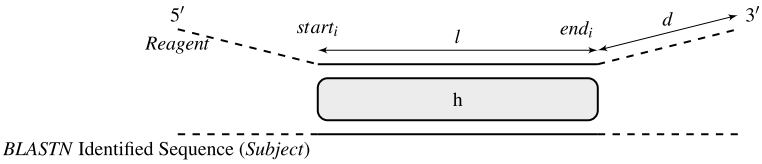


Fig. 2 Notations used to define targeting vs non-targeting reagents. Variable *d* is the distance from the end of the BLASTN-predicted hit to the most 3' nucleotide of the query sequence. Variable *l* is the length of the BLASTN-predicted hit. Variable *h* is the percentage of nucleotides that are identical between the tested reagent (BLASTN *query*) and the identified sequence (BLASTN *subject*). When a hit is composed of several fragments, $start_i$ and end_i are the start and the end of the hit for the *i*th fragment

Table 4 Rule-based classification of BLASTN results produced by the Seek&Blastn fact-checking software detailing the assessment of reagent–gene homology. Red color means that the result found by BLASTN is in contradiction with the claimed status (i.e., targeting, non-targeting). Green color means no contradiction between BLASTN and the claimed status. Orange means questionable status

Case	Sub-case	Claimed status		Case example(s), Error class
		Targeting	Non-Targeting	
Clear hit $(d < 3 \wedge h > 90\% \wedge l > 14) \vee (start_1 = end_2 \vee start_2 = end_1) \vee (l > 16 \wedge h = 100\%)$	$(start_1 = end_2 \vee start_2 = end_1) \vee (d = 0 \wedge h = 100\% \wedge l > 14)$			Targeting shRNA, PCR Primer, Class 0
	$(0 < d < 3) \wedge (90\% < h < 100\%) \wedge (l > 14) \vee (l > 16 \wedge h = 100\%)$			Sub-optimal PCR primer
	$(0 < d < 3) \wedge (90\% < h < 100\%) \wedge (l > 14)$			Questionable PCR primer
	$start_1 = end_2 \vee start_2 = end_1$			Incorrect non-targeting shRNA, Class 7
	$l > 16 \wedge h = 100\%$			Incorrect non-targeting siRNA, Class 7
No clear hit $(h < 90\%) \vee (l < 14) \vee (d > 3 \wedge l < 16) \vee (d > 3 \wedge h < 100\%)$				Incorrect targeting reagent, Class 6; Correct non-targeting shRNA or siRNA, Class 0
BLASTN returns 'No hits found'				Incorrect targeting reagent, Class 6; Non-targeting shRNA or siRNA, Class 0

References

Acuna, D. E., Brookes, P. S., & Kording, K. P. (2018). Bioscience-scale automated detection of figure element reuse. *bioRXiv*. https://doi.org/10.1101/269415.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, *25*(17), 3389–3402. https://doi.org/10.1093/nar/25.17.3389.

Baker, M. (2016). Problematic images found in 4% of biomedical papers [News]. *Nature*,. https://doi.org/10.1038/nature.2016.19802.

Bast, H., & Korzen, C. (2013). The Icecite research paper management system. In *WISE'13: Proceedings of the international conference on web information systems engineering* (Vol. 8181, pp. 396–409). https://doi.org/10.1007/978-3-642-41154-0_30.

Bast, H., & Korzen, C. (2017). A benchmark and evaluation for text extraction from PDF. In *JCDL'17: Proceedings of the 17th ACM/IEEE joint conference on digital libraries* (pp. 99–108). IEEE. https://doi.org/10.1109/jcdl.2017.7991564.

Beck, J. (2011). NISO Z39.96 The Journal Article Tag Suite (JATS): What Happened to the NLM DTDs? *The Journal of Electronic Publishing*. https://doi.org/10.3998/3336451.0014.106.

Brainard, J., & You, J. (2018). What a massive database of retracted papers reveals about science publishing's "death penalty". *Science*,. https://doi.org/10.1126/science.aav8384.

Brischoux, F., & Legagneux, P. (2009). Don't format manuscripts. *The Scientist*, *23*(7), 24.

Bruza, P., & Weeber, M. (2008). *Literature-based discovery* (Vol. 15). Berlin: Springer. https://doi.org/10.1007/978-3-540-68690-3.

Byrne, J. A., & Labbé, C. (2017). Striking similarities between publications from china describing single gene knockdown experiments in human cancer cell lines. *Scientometrics*, *110*(3), 1471–1493. https://doi.org/10.1007/s11192-016-2209-6.

Christopher, J. (2018). Systematic fabrication of scientific images revealed. *FEBS Letters*, *592*(18), 3027–3029. https://doi.org/10.1002/1873-3468.13201.

Citron, D. T., & Ginsparg, P. (2015). Patterns of text reuse in a scientific corpus. *Proceedings of the National Academy of Sciences*, *112*(1), 25–30.

De Bellis, N. (2009). *Bibliometrics and citation analysis: From the Science Citation Index to cybermetrics*. Lanham, MD: Scarecrow Press.

Galea, D., Laponogov, I., & Veselkov, K. (2018). Exploiting and assessing multi-source data for supervised biomedical named entity recognition. *Bioinformatics*, *34*(14), 2474–2482. https://doi.org/10.1093/bioinformatics/bty152.

Gerner, M., Nenadic, G., & Bergman, C. M. (2010). LINNAEUS: A species name identification system for biomedical literature. *BMC Bioinformatics*, *11*(1), 85. https://doi.org/10.1186/1471-2105-11-85.

Giorgi, J. M., & Bader, G. D. (2020). Towards reliable named entity recognition in the biomedical domain. *Bioinformatics*, *36*(1), 280–286. https://doi.org/10.1093/bioinformatics/btz504.

Hartley, J., & Cabanac, G. (2017). The delights, discomforts, and downright furies of the manuscript submission process [Opinion Piece]. *Learned Publishing*, *30*(2), 167–172. https://doi.org/10.1002/leap.1092.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Labbé, C., Grima, N., Gautier, T., Favier, B., & Byrne, J. A. (2019). Semi-automated factchecking of nucleotide sequence reagents in biomedical research publications: The Seek & Blastn tool. *PLoS ONE*, *14*(3), e0213266. https://doi.org/10.1371/journal.pone.0213266.

Labbé, C., & Labbé, D. (2013). Duplicate and fake publications in the scientific literature: how many scigen papers in computer science? *Scientometrics*, *94*(1), 379–396.

Ledford, H., Castelvecchi, D., Dolgin, E., Reardon, S., Gibney, E., Phillips, N., et al. (2017). Nature's 10: Ten people who mattered this year. *Nature*, *552*(7685), 315–324. https://doi.org/10.1038/d41586-017-07763-y.

Liu, X., & Chen, X. (2018). Journal retractions: Some unique features of research misconduct in China. *Journal of Scholarly Publishing*, *49*(3), 305–319. https://doi.org/10.3138/jsp.49.3.02.

Lopez, P. (2009). GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *ECDL'09: Proceedings of the international conference on theory and practice of digital libraries* (Vol. 5714, pp. 473–474). https://doi.org/10.1007/978-3-642-04346-8_62.

Morris, S., Barnas, E., LaFrenier, D., & Reich, M. (2013). The production process. In S. Morris, E. Barnas, D. LaFrenier, & M. Reich (Eds.), *The handbook of journal publishing* (pp. 104–132). Cambridge: Cambridge University Press. https://doi.org/10.1017/cbo9781139107860.00.

Nuijten, M. B., Hartgerink, C. H., van Assen, M. A., Epskamp, S., & Wicherts, J. M. (2016). The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research Methods*, *48*(4), 1205–1226. https://doi.org/10.3758/s13428-015-0664-2.

Pulla, P. (2019). The plan to mine the world's research papers. *Nature*, *571*, 316–318. https://doi.org/10.1038/d41586-019-02142-1.

Smart, P., & Gaston, T. (2019). How prevalent are plagiarized submissions? Global survey of editors. *Learned Publishing*, *32*(1), 47–56. https://doi.org/10.1002/leap.1218.

Soete, L., Schneegans, S., Eröcal, D., Angathevar, B., & Rasiah, R. (2015). A world in search of an effective growth strategy. In S. Schneegans (Ed.), UNESCO science report: Towards 2030 (pp. 20–55). Paris. Retrieved from http://unesdoc.unesco.org/images/0023/002354/235406e.pdf.

Van Noorden, R. (2014). Publishers withdraw more than 120 gibberish papers. *Nature*. https://doi.org/10.1038/nature.2014.14763.

Van Noorden, R. (2015). The image detective who roots out manuscript flaws. *Nature*. https://doi.org/10.1038/nature.2015.17749.

Volentine, R., & Tenopir, C. (2013). Value of academic reading and value of the library in academics' own words. *Aslib Proceedings*, *65*(4), 425–440. https://doi.org/10.1108/AP-03-2012-0025.

Voorhees, E. M. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Communications of the ACM*, *50*(11), 51–54. https://doi.org/10.1145/1297797.1297822.

Yu, J.-Y., DeRuiter, S. L., & Turner, D. L. (2002). RNA interference by expression of shortinterfering RNAs and hairpin RNAs in mammalian cells. *Proceedings of the National Academy of Sciences*, *99*(9), 6047–6052. https://doi.org/10.1073/pnas.092143499.

Affiliations

Cyril Labbé[1] · Guillaume Cabanac[2] · Rachael A. West[3,4] · Thierry Gautier[5] · Bertrand Favier[6] · Jennifer A. Byrne[7,8]

Guillaume Cabanac
guillaume.cabanac@univ-tlse3.fr

Rachael A. West
rachael.west@health.nsw.gov.au

Thierry Gautier
thierry.gautier@univ-grenoble-alpes.fr

Bertrand Favier
bertrand.favier@univ-grenoble-alpes.fr

Jennifer A. Byrne
jennifer.byrne@health.nsw.gov.au

1 CNRS, Grenoble INP, LIG, University of Grenoble Alpes, 38000 Grenoble, France

2 Computer Science Department, IRIT UMR 5505 CNRS, University of Toulouse, 118 route de Narbonne, 31062 Toulouse Cedex 9, France

3 The University of Sydney, Westmead, NSW, Australia

4 Kids Research, Faculty of Medicine and Health, Children's Cancer Research Unit, Westmead, NSW, Australia

5 Centre de Recherche UGA, INSERM U1209/CNRS UMR5309, Institute for Advanced Biology, Site Santé, Allée des Alpes, 38700 La Tronche, France

6 TIMC-IMAG, Team GREPI, University of Grenoble Alpes, Grenoble, France

7 Faculty of Medicine and Health, The University of Sydney, Sydney, NSW, Australia

8 NSW Health Statewide Biobank, Camperdown, NSW, Australia