A Study in Reproducibility: The Congruent Matching Cells Algorithm and cmcR Package

by Joseph Zemmels, Susan VanderPlas, and Heike Hofmann

Abstract Scientific research is driven by our ability to use methods, procedures, and materials from previous studies and further research by adding to it. As the need for computationally-intensive methods to analyze large amounts of data grows, the criteria needed to achieve reproducibility, specifically computational reproducibility, have become more sophisticated. In general, prosaic descriptions of algorithms are not detailed or precise enough to ensure complete reproducibility of a method. Results may be sensitive to conditions not commonly specified in written-word descriptions such as implicit parameter settings or the programming language used. To achieve true computational reproducibility, it is necessary to provide all intermediate data and code used to produce published results. In this paper, we consider a class of algorithms developed to perform firearm evidence identification on cartridge case evidence known as the *Congruent Matching Cells* (CMC) methods. To date, these algorithms have been published as textual descriptions only. We introduce the first open-source implementation of the Congruent Matching Cells methods in the R package **cmcR**. We have structured the **cmcR** package as a set of sequential, modularized functions intended to ease the process of parameter experimentation. We use **cmcR** and a novel variance ratio statistic to explore the CMC methodology and demonstrate how to fill in the gaps when provided with computationally ambiguous descriptions of algorithms.

1 Introduction

Forensic examinations are intended to provide an objective assessment of the probative value of a piece of evidence. Typically, this assessment of probative value is performed by a forensic examiner who visually inspects the evidence to determine whether it matches evidence found on a suspect. The process by which an examiner arrives at their evidentiary conclusion is largely opaque and has been criticized (President's Council of Advisors on Sci. & Tech. 2016) because its subjectivity does not allow for an estimation of error rates. In response, National Research Council (2009) pushed to augment subjective decisions made by forensic examiners with automatic algorithms that objectively assess evidence and can be explained during court testimony. In addition to the objectivity of these algorithms, there is an additional benefit: we expect that an algorithm with the same random seed run on the same data multiple times will produce the same answer; that is, that the results are repeatable. This is extremely beneficial because it allows the prosecution and defense to come to the same conclusion given objective evidence or data.

Repeatability and reproducibility

Repeatability in forensic labs is enforced primarily using standard operating procedures (SOPs), which specify the steps taken for any given evaluation, along with the concentrations of any chemicals used, the range of acceptable machine settings, and any calibration procedures required to be completed before the evidence is evaluated. When labs use computational procedures, this SOP is augmented with specific algorithms, which are themselves SOPs intended for use by man and machine. Algorithms are generally described on two levels: we need both the conceptual description (intended for the human using the algorithm) and the procedural definition (which provides the computer hardware with a precise set of instructions). For scientific and forensic repeatability and reproducibility, it is essential to have both pieces: the algorithm description is critical for establishing human understanding and justifying the method's use in court, but no less important is the computer code which provides the higher degree of precision necessary to ensure the results obtained are similar no matter who evaluates the evidence. As with SOPs in lab settings, the code parameters function like specific chemical concentrations; without those details, the SOP would be incomplete and the results produced would be too variable to be accepted in court.

The National Academy of Sciences, Engineering, and Medicine (2019) defines *reproducibility* as "obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis." This form of reproducibility requires that the input data, code, method, and computational environment are all described and made available to the community. In many situations, this level of reproducibility is not provided – not just in forensics but in many other

applied disciplines. In forensics in particular, it is easier to list the exceptions: reproducible algorithms have been proposed in sub-disciplines including DNA (Tvedebrink, Andersen, and Curran 2020; Goor, Hoffman, and Riley 2020; Tyner et al. 2019), glass (Curran, Champod, and Buckleton 2000; Park and Tyner 2019), handwriting (Crawford 2020), shoe prints (Park and Carriquiry 2020), and ballistic evidence (Hare, Hofmann, and Carriquiry 2017; Tai and Eddy 2018).

We find it useful to instead consider a more inclusive hierarchy of reproducibility. Algorithms at higher tiers of the hierarchy are more easily reproducible in the sense that fewer resources are required to (re)-implement the algorithm.

Definition 1 *Hierarchy of Reproducibility*

Conceptual description *The algorithm is described and demonstrated in a scientific publication.*

Pseudocode *The algorithm is described at a high level of detail with pseudocode implementation provided, and results are demonstrated in a scientific publication.*

Reproducible data *The algorithm is described and demonstrated in a scientific publication, and input data are available in supplementary material.*

Comparable results *The algorithm is described and demonstrated in a scientific publication, and input data and numerical results are provided in supplementary material.*

Full reproducibility *The algorithm is described and demonstrated in a scientific publication, and the input data, source code, parameter settings, and numerical results are provided in supplementary material.*

To aid in comprehension of an algorithm, it is useful to supplement conceptual descriptions with pseudocode. However, a conceptual description and pseudocode alone do not contain sufficient detail (e.g., parameter settings) to ensure computational reproducibility. Implementing algorithms based on conceptual descriptions or pseudocode requires enumerating and testing possible parameter choices which, depending on their complexity, can be a lengthy and expensive process. In contrast, implementing fully reproducible algorithms requires only as much time as it takes to emulate the original development environment. Commonly identified reasons for unreproducible results include (1) ambiguity in how procedures were implemented, (2) missing or incomplete data, and (3) missing or incomplete computer code to replicate all statistical analyses (Leek and Jager 2017). In particular, for statistical algorithms which depend on input data, we find that full reproducibility depends on the provision of both original data and any manual pre-processing applied to said data, as this manual process is not reproducible by itself. In combination with the code, the algorithm description, and the numerical results presented in the paper, it should be possible to fully reproduce the results of a paper.

In this paper, we demonstrate the importance of higher levels of reproducibility by examining the Congruent Matching Cells (CMC) algorithm for cartridge case comparisons and developing an open-source, fully reproducible version for general use in the forensics community.

The Congruent Matching Cells algorithm

A *cartridge case* is the portion of firearm ammunition that encases a projectile (e.g., bullet, shots, or slug) along with the explosive used to propel the projectile through the firearm. When a firearm is discharged, the projectile is propelled down the barrel of the firearm, while the cartridge case is forced towards the back of the barrel. It strikes the back wall, known as the *breech face*, of the barrel with considerable force, thereby imprinting any markings on the breech face onto the cartridge case and creating the so-called *breech face impressions*. These markings are used in forensic examinations to determine whether two cartridge cases have been fired by the same firearm. During a forensic examination, two pieces of ballistic evidence are placed under a *comparison microscope*. Comparison microscopes allow for a side-by-side comparison of two objects within the same viewfinder, as seen in Figure 1. A pair of breech face images is aligned along the thin black line in the middle of the images. The degree to which these breech face markings can be aligned is used to determine whether the two cartridge cases came from the same source; i.e., were fired from the same firearm. These breech face impressions are considered analogous to a firearm's "fingerprint" left on a cartridge case (Thompson 2017).

The Congruent Matching Cells (CMC) pipeline is a collection of algorithms to process and compare cartridge case evidence (Song 2013). Since its introduction, the pipeline and its extensions (Tong, Song, and Chu 2015; Chen et al. 2017; Song et al. 2018) have shown promise in being able to differentiate between matching and non-matching cartridge cases. However, so far the CMC pipelines have only been introduced in the form of conceptual descriptions. Further, the cartridge case scans used to validate the pipelines are only available in their raw, unprocessed forms on the NIST Ballistics

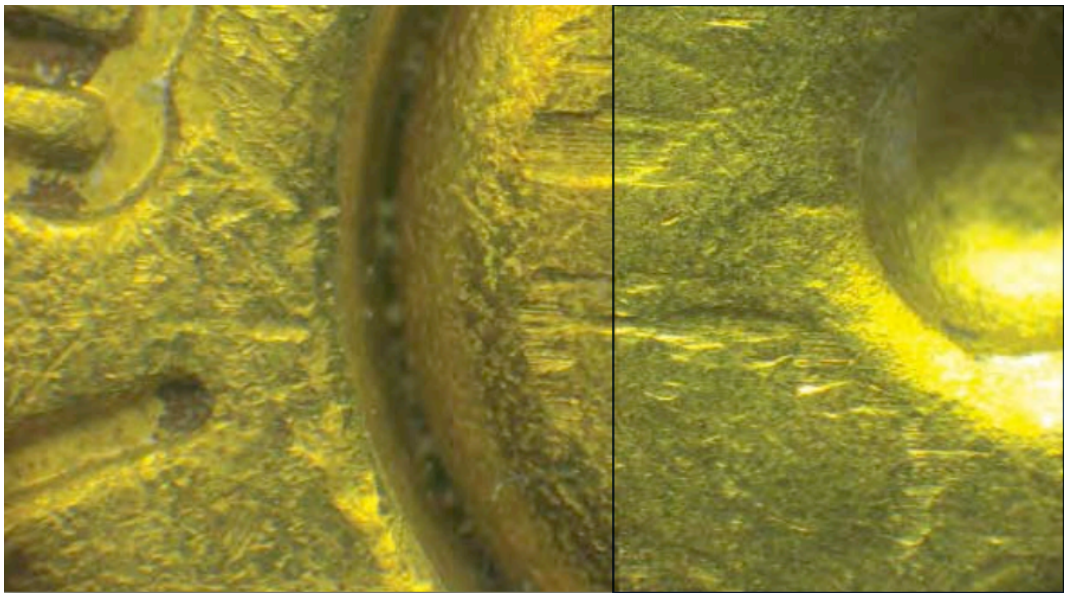


Figure 1: A cartridge case pair with visible breech face impressions under a microscope. A thin line can be seen separating the two views. The degree to which the markings coincide is used to conclude whether the pair comes from the same source.

Toolmark Research Database (Zheng, Soons, and Thompson 2016). While it is clear that the creators of the CMC pipeline have a working implementation, the wider forensic science community only has access to conceptual descriptions of the pipeline and summary statistics describing its performance. In our hierarchy of reproducibility, this puts the CMC algorithm somewhere between the conceptual description and reproducible data stage: the steps are described but no code is available, and the raw data are available but manual pre-processing steps make this raw data insufficient to replicate the pipeline even with newly written code.

The development of the CMC algorithm seems to be representative of how many forensic algorithms are developed: after an algorithm is introduced, researchers build upon the foundation laid by the original algorithm in subsequent papers. These changes are often incremental in nature and reflect a growing understanding of the algorithm's behavior. While this cycle of scientific progress certainly is not unique to forensic algorithms, given the gravity of the application it is imperative that these incremental improvements not be unnecessarily delayed. As such, we believe that the forensic community at-large would benefit greatly by establishing an open-source foundation for their algorithms upon which additional improvements can be developed. Using open-source algorithms are cheaper to use than writing one's own code, enables the process of peer review by providing an accessible benchmark, and helps other research groups or companies stay on the leading edge of technology development (The Linux Foundation 2017).

Here, we describe the process of implementing the CMC pipeline for the comparison of marks on spent cartridge cases, using the descriptions from two published papers, Song et al. (2014) and Tong, Song, and Chu (2015). Our R package, **cmcR**, provides an open-source implementation of the CMC pipeline. We use **cmcR** to illustrate how ambiguities in the textual description of an algorithm can lead to highly divergent results. In particular, our implementation highlights an extreme sensitivity to processing and parameter decisions that has not been discussed previously. Additionally, we argue that our implementation can be used as a template for future implementations of forensic pattern-matching algorithms to not only ensure transparency and auditability, but also to facilitate incremental improvements in forensic algorithms.

In the remainder of this paper, we describe a general, reproducible, and open-source CMC pipeline which encompasses those discussed in Song (2013), Song et al. (2014), and Tong, Song, and Chu (2015). Song (2013) lays out the conceptual framework for the original CMC pipeline later implemented in Song et al. (2014) and Tong et al. (2014). An improvement of the pipeline presented in Tong, Song, and Chu (2015) and used in subsequent papers is referred to as the "High CMC" method (Chen et al. 2017). However, it should be noted that what the authors refer to as the original and High CMC decision rules are variations of one step of a larger CMC pipeline.

The **cmcR** package contains implementations designed for use with 3D topographical scans of the original decision rule described in Song (2013) and Song et al. (2014) and the High CMC decision rule described in Tong, Song, and Chu (2015). The source code to the full **cmcR** package is accessible at

https://github.com/CSAFE-ISU/cmcR.

2 The CMC pipeline

In this section, we examine the process of implementing the CMC pipeline for automatic comparisons of 3D cartridge case scans. At each step, we will discuss how we filled in the gaps of the original description during the creation of **cmcR**.

All of the CMC pipelines can be broken down into three broad stages: (1) pre-processing, (2) cell-based similarity feature extraction, and (3) application of a decision rule as illustrated in Figure 2. In the following sections we break each of these stages further into a set of modular steps. One advantage of modularizing these algorithms is that we can implement an algorithm as a set of sequential procedures. This allows us to test new variations against the old implementation in a coherent, unified framework.

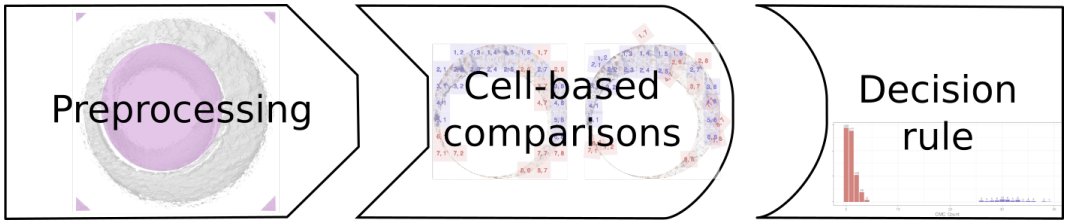


Figure 2: The stages of CMC pipelines. In the pre-processing stage, each scan is prepared for analysis, removing extraneous information and noise. Then, each scan is broken up into cells, which are numerically compared to cells in the other scan to determine an optimal alignment. Finally, each of the scores arising from the cells in the second stage are compared to a reference distribution to determine whether the scans originate from the same source or from different sources.

The primary difference between the two pipelines presented here, using the original and High CMC decision rules, lies in how the decision rules are utilized to separate matching vs. non-matching cartridge case pairs. In addition, there are also several small differences in the parameters used in the pre-processing and comparison procedures.

Initial data

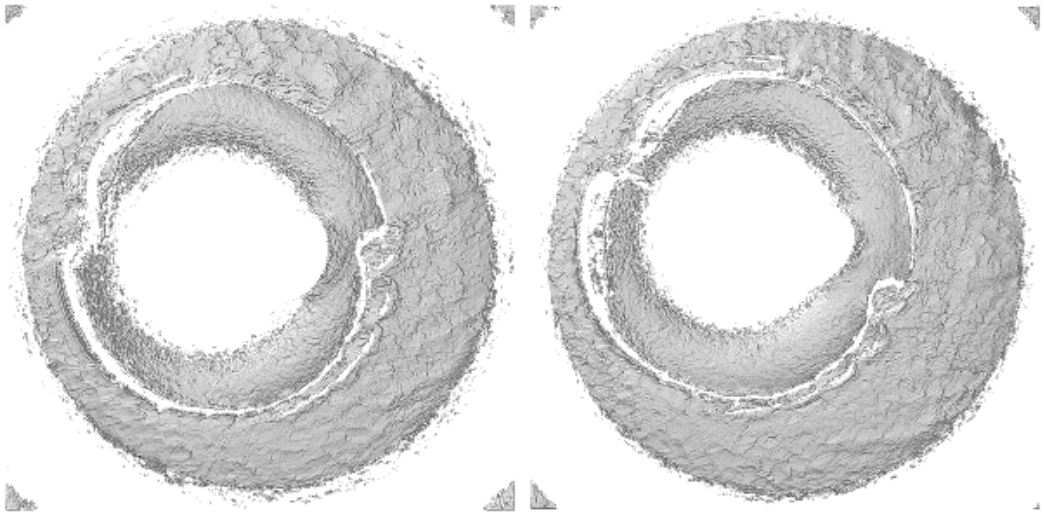
Digital microscopy is capable of precision measurements of surface topology at high resolutions. Using a 3D microscope, we can obtain scans of breech face impressions at the micron level ($1\mu m = 10^{-3}mm = 10^{-6}m$). These 3D topological scans are used as input to automated comparison algorithms, such as the CMC pipeline originally proposed in Song (2013). We will use the same data set referenced in Song et al. (2014) and Tong, Song, and Chu (2015) to illustrate usage of the **cmcR** package. These 3D scans of cartridge cases are available from the NIST Ballistics Toolmark Research Database (Zheng, Soons, and Thompson 2016). The strings defined below refer to three cartridge case scans available on the NBTRD from Fadul et al. (2011) and will be used throughout the remainder of this paper.

```
library(cmcR)

nbtrd_url <- "https://tsapps.nist.gov/NRBTD/Studies/CartridgeMeasurement"

x3p_ids <- c("DownloadMeasurement/2d9cc51f-6f66-40a0-973a-a9292dbee36d",
             "DownloadMeasurement/cb296c98-39f5-46eb-abff-320a2f5568e8",
             "DownloadMeasurement/8ae0b86d-210a-41fd-ad75-8212f9522f96")

file_names <- c("fadul1-1.x3p","fadul1-2.x3p","fadul2-1.x3p")

purrr::walk2(.x = x3p_ids,
             .y = file_names,
             .f = function(x3p_id,file_name){
               download.file(url = file.path(nbtrd_url, x3p_id),
                             destfile = paste0("data/",file_name),mode = "wb")
             })
```

Cartridge case scans are commonly stored in the ISO standard x3p file format ("Geometrical product specifications (GPS) — Surface texture: Areal — Part 72: XML file format x3p" 2017). x3p

is a container format which consists of a single surface matrix representing the height value of the breech face surface and metadata concerning the parameters under which the scan was taken (size, resolution, creator, microscope, microscopy software versions, etc.). The **x3ptools** package provides functionality to work with the format in R (Hofmann et al. 2020).

Figure 3 shows the surface matrices of a known match (KM) pair of cartridge cases from a study by Fadul et al. (2011). In this study, a total of 40 cartridge cases were scanned with a lateral resolution of 6.25 microns (micrometers) per pixel. The surface matrices are approximately 1200×1200 pixels in size corresponding to an area of about 3.8×3.8 mm^2.



(a) (b)

Figure 3: Unprocessed surface matrices of the known-match Fadul 1-1 and Fadul 1-2 Fadul et al. 2011. The observations in the corners of these surface matrices are artifacts of the staging area in which these scans were taken. The holes on the interior of the primer surfaces are caused by the firing pin striking the primer during the firing process. The region of the primer around this hole does not come into uniform contact with the breech face of the firearm.

Only certain regions of a cartridge case contain identifying breech face impression markings. Song (2013) defines "valid correlation regions" as regions where "the individual characteristics of the ballistics signature are found that can be used effectively for ballistics identification." Prior to applying the CMC comparison procedure, cartridge scans must undergo some pre-processing to isolate the valid correlation regions.

Pre-processing procedures

During the pre-processing stage, we apply sequential steps to prepare each cartridge case for analysis. The goal of this process is to remove the edges and center of the scan which did not come into contact with the breech face, as well as any artifacts of the scan and microscope staging which do not accurately represent the breech face surface. The various iterations of the CMC algorithm describe different variations of these steps. A summary of these steps is shown in Figure 4.

Translating the pre-processing steps in Figure 4 into an implementation requires the implementer to decide between potentially many implicit parameter choices. For example, Table 1 compares the pre-processing procedures as described in Song et al. (2014) to considerations that need to be made when implementing the procedures. Depending on one's interpretation of the description, there are many possible implementations that satisfy the described procedure - in contrast, there was only one implementation that led to the original results. While not explicitly mentioned in Song et al. (2014), Song et al. (2018) indicates that the "trimming" of the unwanted regions of the scan is performed manually. It is difficult to replicate manual steps as part of a reproducible pipeline; the best solution is for the authors to provide intermediate data after the manual steps have been completed.

The pre-processing procedures are implemented via modularized functions of the form `preProcess_*`. Modularizing the steps of the pre-processing procedures makes the overall process easier to understand and allows for experimentation. Figure 5 shows an overview of the pre-processing framework for the Fadul 1-1 breech face from reading the scan (left) to an analysis-ready region (right). For each scan in Figure 5, eleven height value percentiles: the Minimum (0th), 1st, 2.5th, 10th, 25th, Median

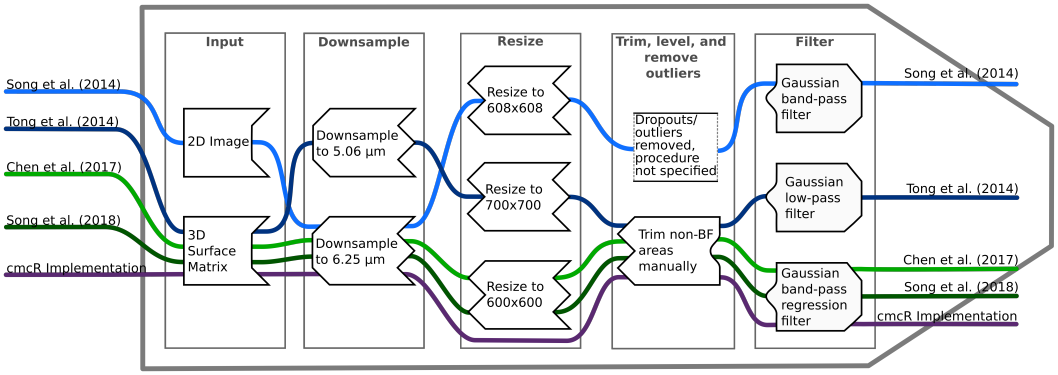


Figure 4: Overview of the set of pre-processing steps used in the CMC algorithms. Where a procedure step is not discussed or explicitly not applied in the paper, the path traverses empty space.

Table 1: Description of pre-processing procedures from Song et al. 2014 vs. considerations that need to be made when implementing these procedures. Each of these considerations requires the implementer to decide between potentially many choices.

Description from Song et al. (2014)	**Implementation Considerations**
"Trim off the inside firing pin surface and other areas outside the breech face mark, so that only breech face impression data remain for correlation."	Removal of firing pin hole, primer exterior, global trend, and primer roll-off
"Identify and remove dropouts or outliers."	Definition of outliers, what "removal" of dropouts or outliers means
"Apply a band-pass Gaussian regression filter with 40 μm short cut-off length and 400 μm long cut-off length to remove low frequency components, including surface curvature, form error, waviness and high frequency components which mainly arise from the instrument noise."	Wavelength cut-off parameters, specific implementation of the filter

(50th), 75th, 90th, 97.5th, 99th, and Maximum (100th) are mapped to a purple-to-orange color gradient. This mapping is chosen to highlight the extreme values in each scan.

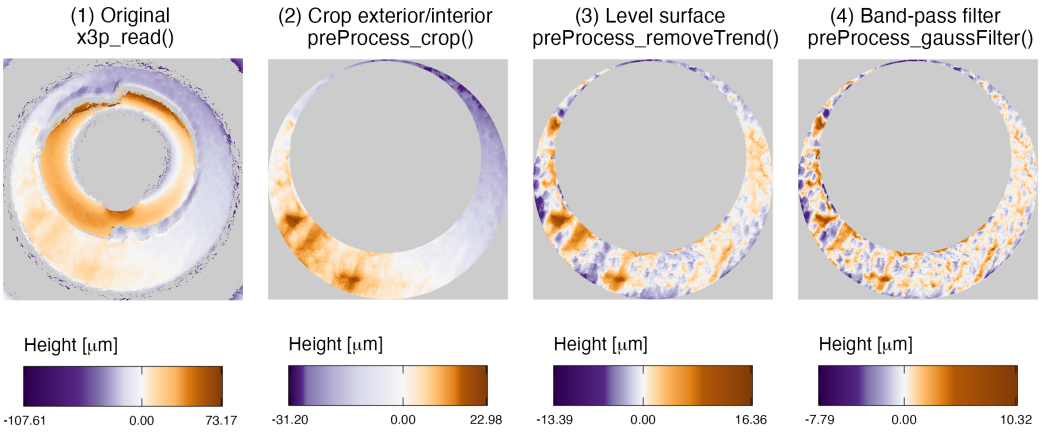


Figure 5: Illustration of the sequential application of pre-processing steps implemented in **cmcR**. We map the cartridge case surface height values to a divergent purple-white-orange color scale to emphasize deviations from the median height value (represented here as 0 micrometers). At each stage, the variability in height across the scan decreases as we emphasize the regions containing breech face impressions.

We demonstrate usage of the `preProcess_*` functions on the Fadul 1-1 scan. Each code chunk is followed up with an explanation of the functions used.

```
# Step (1)
fadul1.1 <- x3ptools::x3p_read("data/fadul1-1.x3p")
```

We begin with a 3D scan. Typically, we downsample scans to about 25% of their size by only retaining every other row and column in the surface matrix. The breech faces in Fadul et al. (2011) were initially scanned at a resolution of 3.125 μm per pixel. Downsampling reduces the resolution to 6.25 μm per pixel. Step (1) in Figure 5 shows an unprocessed breech face scan.

```
# Step (2)
fadul1.1_cropped <- fadul1.1 %>%
  cmcR::preProcess_crop(region = "exterior") %>%
  cmcR::preProcess_crop(region = "interior")
```

We then use a labeling algorithm to identify three major regions of the scan: the exterior of the cartridge case primer, the breech face impression region of interest, and the firing pin impression region in the center of the scan (Hesselink, Meijster, and Bron 2001; Barthelme 2019). We remove observations outside of the breech face impression region (i.e., replaced with `NA`). The resulting breech face scan, like the one shown in step (2) of Figure 5, is reproducible assuming the same parameters are used. The `preProcess_crop` function removes the exterior and firing pin impression region on the interior based on the `region` argument.

```
# Step (3)
fadul1.1_deTrended <- fadul1.1_cropped %>%
  preProcess_removeTrend(statistic = "quantile", tau = .5, method = "fn")
```

In step (3), we remove the southwest-to-northeast trend observable in steps (1) and (2) of Figure 5 by subtracting the estimated conditional median height value. The result of the `preProcess_removeTrend` function the median-leveled breech face scan in step (3) of Figure 5.

```
# Step (4)
fadul1.1_processed <- fadul1.1_deTrended %>%
  preProcess_gaussFilter(filtertype = "bp", wavelength = c(16,500)) %>%
  x3ptools::x3p_sample(m = 2)
```

Finally, we apply a band-pass Gaussian filter to the surface values to attenuate noise and unwanted large-scale structure. Step (4) of Figure 5 shows the effect of the `preProcess_gaussFilter` function. There is currently no determination or removal of outliers in the **cmcR** package's pre-processing procedures. Instead, we rely on the low-pass portion of the Gaussian filter to reduce the effects of any high-frequency noise.

Figure 6 displays the processed Fadul 1-1 and Fadul 1-2 scans; the second matrix is processed using the same parameters. Next, similarity features are extracted from a processed cartridge case pair in the cell-based comparison procedure.

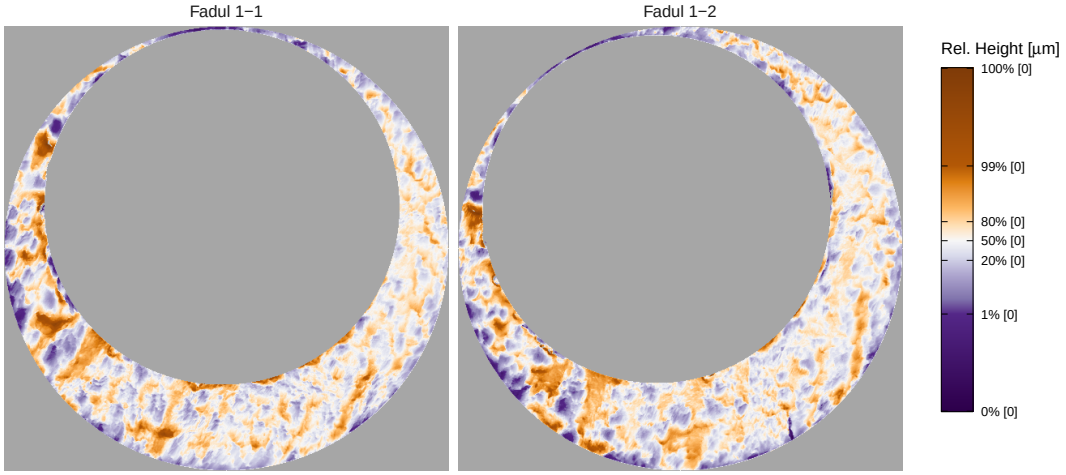


Figure 6: Fadul 1-1 and Fadul 1-2 after pre-processing. Similar striated markings are now easier to visually identify on both surfaces. It is now clearer that one of the scans needs to be rotated to align better with the other.

"Correlation cell" comparison procedure

As described in Song (2013), breech face markings are not uniformly impressed upon a cartridge case during the firing process. As such, only certain sections of the cartridge case are used in a comparison. In the CMC pipeline as proposed by Song (2013) two scans are compared by partitioning one breech face scan into a grid of so-called "correlation cells". These cells are compared individually to their best-matching counterpart on the other scan. If a large proportion of these correlation cells are highly similar to their counterparts on the other breech face scan, this is considered as evidence that the markings on the two cartridge cases were made by the same source. The number of highly similar cells is defined as the *CMC count C* (Song 2013) of the breech-face comparison. The CMC count is considered to be a more robust measure of similarity than the correlation calculated between two full scans.

Figure 7 illustrates the cell-based comparison procedure between two cartridge case scans. The scan on the left serves as the reference; it is divided into a grid of 8×8 cells.

Figure 8 shows the steps of the correlation cell comparison process in each of the papers as well as the **cmcR** implementation. Each cell is paired with an associated larger region in the other scan. The absolute location of each cell and region in their respective surface matrices remain constant. However, the scan on the right is rotated to determine the rotation at which the two scans are the most "similar," as quantified by the *cross-correlation function* (CCF).

For real-valued matrices A and B of dimension $M \times N$ and $P \times Q$, respectively, the cross-correlation function, denoted $(A \star B)$ is defined as

$$(A \star B)[m,n] = \sum_{i=1}^{M} \sum_{j=1}^{N} A[i,j]B[(i+m),(j+n)],$$

where $1 \leq m \leq M+P-1$ and $1 \leq n \leq N+Q-1$. By this definition, the $[m,n]$th element of the resulting $M+P-1 \times N+Q-1$ CCF matrix quantifies the similarity between matrices A and B for a translation of matrix B by m pixels horizontally and n pixel vertically. The index at which the CCF attains a maximum represents the optimal translation needed to align B with A. The CCF as

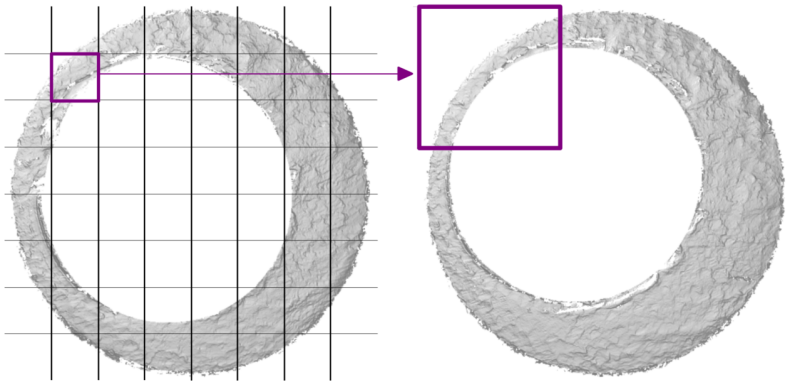


Figure 7: Illustration of comparing a cell in the reference cartridge case scan (left) to a larger region in a questioned cartridge case scan (right). Every one of the cells in the reference cartridge case is similarly paired with a region in the questioned cartridge case. To determine the rotation at which the two cartridge cases align, the cell-region pairs are compared for various rotations of the questioned cartridge case.

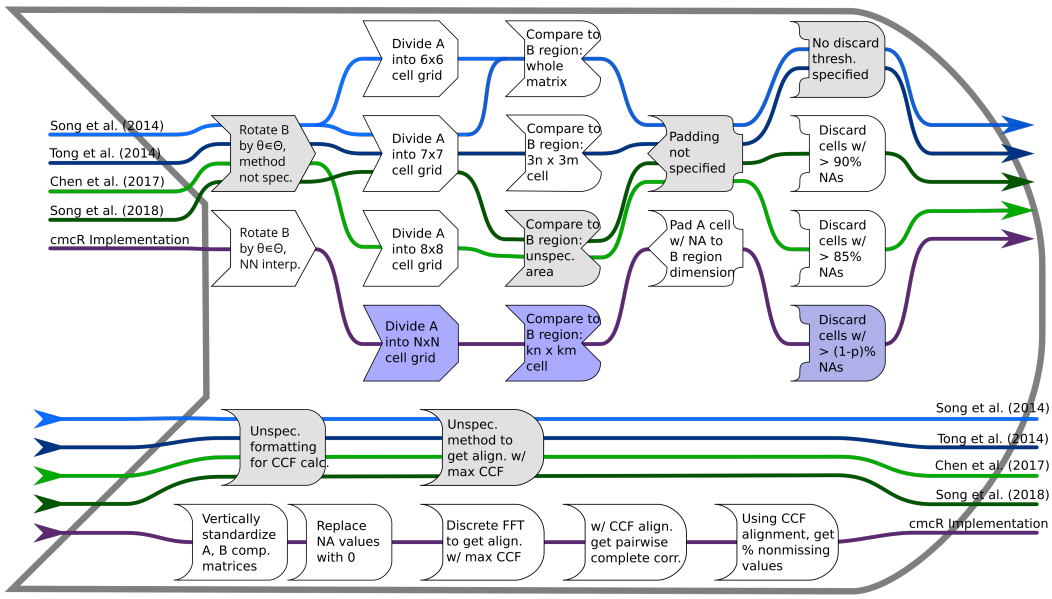


Figure 8: Each CMC implementation uses a slightly different procedure to obtain a similarity score between two cartridge cases. Steps which are implemented with additional user-specified parameters are shaded purple; steps which are described but without sufficient detail are shaded grey.

defined need not be bounded between -1 and 1. However, it is common to normalize the CCF for interpretability, and this is the convention adopted in the **cmcR** package.

Prior to calculating the CCF, the matrices A and B are standardized through subtraction of their respective means and division by their respective standard deviations. This is referred to as the *Areal Cross-Correlation Function* (ACCF) in some CMC papers (Ott, Thompson, and Song 2017). A direct calculation of the CCF for breech face scans based on the definition above is prohibitively slow. While computationally feasible alternatives exist, Song (2013) and other CMC papers do not specify the algorithm used to calculate the CCF.

Published descriptions of the CMC algorithm do not detail how the CCF is calculated. In image processing, it is common to use an implementation based on the Fast Fourier Transform (Brown 1992). This implementation leverages the Cross-Correlation Theorem, which states that for matrices A and B, the CCF can be expressed in terms of a frequency-domain pointwise product:

$$(A \star B)[m,n] = \mathcal{F}^{-1}\left(\overline{\mathcal{F}(A)} \odot \mathcal{F}(B)\right)[m,n],$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the discrete Fourier and inverse discrete Fourier transforms, respectively, and $\overline{\mathcal{F}(A)}$ denotes the complex conjugate (Brigham 1988). Because the product on the right-hand side is calculated pointwise, we trade the moving sum computations from the definition of the CCF for two forward Fourier transformations, a pointwise product, and an inverse Fourier transformation. The Fast Fourier Transform (FFT) algorithm can be used to reduce the computational load considerably. Our implementation of this FFT-based CCF calculation is adapted from the **cartridges3D** package (Tai 2021).

No computational shortcut comes without some trade-offs, though, and this FFT-based CCF calculation is no different. The FFT does not tolerate missing values, and breech faces are not continuous surfaces – all of the white regions in Figure 7 correspond to missing values. While it is unclear how the CCF is implemented in the CMC papers, the **cmcR** package adopts the following conventions:

- Only cells with a minimum proportion of non-missing pixels are assessed. This minimum threshold differs across CMC papers (15% in Chen et al. (2017) vs. 10% in Song et al. (2018), as shown in Figure 8), and is referenced but not specified in several other papers (Tong et al. 2014; Song et al. 2014; Chu, Tong, and Song 2013). The `comparison_calcPropMissing` function computes the proportion of a matrix that is missing (`NA`-valued).
- Missing values are replaced with the overall mean value when the FFT-based CCF is computed (using function `comparison_replaceMissing`).
- The optimal translation is determined using the FFT-based CCF (using `comparison_fft_ccf`).
- Based on the optimal translation determined from the FFT-based CCF, we compute the pairwise complete CCF directly, avoiding any distortion of the CCF computation based on compensation for missing values (using function `comparison_cor`).

All of the steps dealing with cell-based comparisons are implemented as functions of the form `comparison_*`. Similar to the `preProcess_*` functions, the `comparison_*` functions can be chained together through a sequence of pipes. Below, we use the `comparison_allTogether` function to perform the entire cell-based comparison procedure in one call. The comparison procedure is performed twice: once with Fadul 1-1 considered the "reference" scan divided into cells that are compared to the "target" scan Fadul 1-2 and again with the roles reversed.

```
# Fill in most of the arguments first
comp_w_pars <- purrr::partial(.f = comparison_allTogether,
                              numCells = c(8,8), maxMissingProp = .85)

# Then, map the remaining values to theta
kmComparisonFeatures <- purrr::map_dfr(
  seq(-30,30,by = 3),
  ~comp_w_pars(reference = fadul1.1, target = fadul1.2, theta = .))

kmComparisonFeatures_rev <- purrr::map_dfr(
  seq(-30,30,by = 3),
  ~comp_w_pars(reference = fadul1.2, target = fadul1.1, theta = .))
```

The `comparison_allTogether` function consists of the following steps wrapped into a single convenience function:

Table 2: Example of output from correlation cell comparison procedure between Fadul 1-1 and Fadul 1-2 rotated by -24 degrees. Due to the large proportion of missing values that are replaced to compute the FFT-based correlation, the pairwise-complete correlation is most often greater than the FFT-based correlation.

Cell Index	Pairwise-comp. corr.	FFT-based corr.	Δx	Δy	θ
2, 7	0.432	0.228	-14	-33	-24
2, 8	0.464	0.176	9	-44	-24
3, 1	0.841	0.478	-7	15	-24
3, 8	0.699	0.277	-7	5	-24
4, 1	0.850	0.375	-4	11	-24

- `comparison_cellDivision`: Divide the reference scan into cells
- `comparison_getTargetRegions`: Extract regions associated with each reference cell from the target scan
- `comparison_calcPropMissing`: Compute missing proportions and filter out cells with a proportion of missing values above the threshold.
- `comparison_standardizeHeights`: Standardize height values
- `comparison_replaceMissing`: Replace missing values
- `comparison_fft_ccf`: Compute CCF and estimated translations using FFT
- `comparison_alignedTargetCell`: Extract a matrix from the target scan corresponding to the region of the target scan to which the reference cell aligns
- `cor`: Calculate the pairwise-complete correlation between each cell pair

The `comparison_allTogether` is called repeatedly while rotating the target scan by a set of rotation angles. When implementing the High CMC decision rule (Tong, Song, and Chu 2015), both combinations of reference and target scan are examined (e.g. A-B and B-A).

Table 2 shows several rows of the data frame output of the `comparison_allTogether` function for the comparison of Fadul 1-1 vs. Fadul 1-2 considering Fadul 1-1 as the reference scan. Although we used a grid of 8×8 cells, there were only 26 cell-region pairs that contained a sufficient proportion of non-missing values (15% in this example). The features derived from the correlation cell procedure (CCF_{max}, Δx, Δy, θ) are then used to measure the similarity between scans.

Decision rule

For each cell on the reference scan, we calculate the translation $(\Delta x, \Delta y)$ and cross-correlation across rotations by a set of angles θ of the target scan. The task is to determine whether multiple cells come to a "consensus" on a particular translation and rotation. If such a consensus is reached, then there is evidence that a true aligning translation and rotation exists and the cartridge cases match. The CMC decision rules principally differ in how they identify consensus among the $\Delta x, \Delta y, \theta$ values. Here, we describe the two pipelines implemented in the **cmcR** package: using the original decision rule described in Song et al. (2014) and the High CMC decision rule proposed in Tong, Song, and Chu (2015).

The Original CMC decision rule

This section briefly describes the decision rule used in the first CMC paper (Song 2013). For a thorough explanation of the procedure, refer to the CMC Decision Rule Description vignette of the **cmcR** package.

Let x_i, y_i, θ_i denote the translation and rotation parameters which produce the highest CCF for the alignment of cell-region pair i, $i = 1, ..., n$ where n is the total number of cell-region pairs containing a sufficient proportion of non-missing values. Song (2013) propose the median as a consensus $(x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}})$ across the cell-region pairs. Then, the distance between each (x_i, y_i, θ_i) and

Table 3: Different thresholds for translation, rotation, and CCF_{max} are used across different papers. The range in CCF_{max} is particularly notable.

Paper	**Translation T_x, T_y (in pixels)**	**Rotation θ (in degrees)**	CCF_{max}
Song et al. (2014)	20	6	0.60
Tong et al. (2014)	30	3	0.25
Tong et al. (2015)	15	3	0.55
Chen et al. (2017)	20	3	0.40
Song et al. (2018)	20	6	0.50

$(x_{\text{ref}}, y_{\text{ref}}, \theta_{\text{ref}})$ is compared to thresholds $T_x, T_y, T_\theta, T_{\text{CCF}}$. A cell-region pair i is declared a "match" if all of the following conditions hold:

$$\begin{aligned} |x_i - x_{\text{ref}}| &\leq T_x, \\ |y_i - y_{\text{ref}}| &\leq T_y, \\ |\theta_i - \theta_{\text{ref}}| &\leq T_\theta, \\ \text{CCF}_{\text{max},i} &\geq T_{\text{CCF}}. \end{aligned} \tag{1}$$

The number of matching cell-region pairs, the "CMC count," is used as a measure of similarity between the two cartridge cases. Song et al. (2014) indicate that the thresholds $T_x, T_y, T_\theta, T_{\text{CCF}}$ need to be determined experimentally. Table 3 summarizes the thresholds used in various CMC papers.

Unlike the original CMC pipeline, the High CMC decision rule considers multiple rotations for each cell-region pair.

The High CMC decision rule

For the High CMC decision rule, two scans are compared in both directions - i.e., each scan takes on the role of the reference scan that is partitioned into a grid of cells. Tong, Song, and Chu (2015) claim that some matching cell-region pairs "may be mistakenly excluded from the CMC count" under the original decision rule because they attain the largest CCF at a rotation outside the range allowed by T_θ "by chance."

Tong, Song, and Chu (2015) introduce consensus values across all cell-region pairs for each rotation angle θ and calculate a θ-dependent CMC count as the sum of matches observed. Under the High CMC rule, a cell-region pair i is defined as a match conditional on a particular rotation θ if it satisfies the following three conditions:

$$\begin{aligned} |x_{i,\theta} - x_{ref,\theta}| &\leq T_x \\ |y_{i,\theta} - y_{ref,\theta}| &\leq T_y \\ \text{CCF}_{i,\theta} &\geq T_{\text{CCF}}. \end{aligned} \tag{2}$$

The θ-dependent CMC count, CMC_θ, is defined as the sum of matching cell-region pairs.

Tong, Song, and Chu (2015) assert that for a truly matching cartridge case pair, the relationship between θ and CMC_θ should exhibit a "prominent peak" near the true rotation value. That is, CMC_θ should be largest when the scans are close to being correctly aligned. Further, non-matching pairs should exhibit a "relatively flat and random [...] pattern" across the CMC_θ values.

To determine whether a "prominent peak" exists in the relationship between θ and CMC_θ, Tong, Song, and Chu (2015) consider an interval of rotation angles with large associated CMC_θ values. Let $CMC_{\max} = \max_\theta CMC_\theta$ be the maximum CMC_θ count across all rotation angles. For $\tau > 0$, define $S(\tau) = \{\theta : CMC_\theta > (CMC_{\max} - \tau)\}$ as the set of rotations with "large" CMC_θ values. Tong, Song,

and Chu (2015) consider the "angular range" as $R(\tau) = |\max_\theta S(\tau) - \min_\theta S(\tau)|$. If $R(\tau)$ is small, then there is evidence that many cells agree on a single rotation and that the scans match. To arrive at a CMC count similarity score, Tong, Song, and Chu (2015) suggest a value for τ of 1 and determine:

> If the angular range of the "high CMCs" is within the range T_θ, identify the CMCs for each rotation angle in this range and combine them to give the number of CMCs for this comparison in place of the original CMC number.

If the angular range is larger than T_θ, we say that the cartridge case pair "fails" the High CMC criteria and the original CMC number is used. The High CMC decision rule returns a CMC count at least as large as the original decision rule.

Implementation of decision rules

In this section, we implement the decision rules in **cmcR** for both the original and High CMC decision rules. For illustrative purposes, we consider a set of thresholds: $T_x = T_y = 20$, $T_\theta = 6$, and $T_{\text{CCF}} = 0.5$.

Decision rules in **cmcR** are implemented as functions of the form `decision_*`. In particular, the `decision_CMC` function applies both the original and High CMC decision rules depending on if the parameter τ is set. The code below demonstrates the use of `decision_CMC` on the features `kmComparisonFeatures`, extracted from the comparison of scans Fadul 1-1 vs. Fadul 1-2. Convserely, `kmComparisonFeatures_rev` contains the features from a comparison of Fadul 1-2 vs. Fadul 1-1. For comparison, we also compute the CMCs under both decision rules for the comparison between the non-match pair Fadul 1-1 and Fadul 2-1 (not shown to avoid redundancy).

```
kmComparison_cmcs <- kmComparisonFeatures %>% mutate(
  originalMethodClassif =
    decision_CMC(cellIndex = cellIndex, x = x, y = y, theta = theta,
                 corr = pairwiseCompCor, xThresh = 20, thetaThresh = 6,
                 corrThresh = .5),
  highCMCClassif =
    decision_CMC(cellIndex = cellIndex, x = x, y = y, theta = theta,
                 corr = pairwiseCompCor, xThresh = 20, thetaThresh = 6,
                 corrThresh = .5, tau = 1))
```

We use the `cmcPlot` function to visualize congruent matching cells (CMCs) and non-congruent matching cells (non-CMCs). Figure 9 shows the CMCs and non-CMCs in blue and red, respectively, based on the original decision rule. The (red) non-CMC patches are shown in the position where the maximum CCF value in the target scan is attained. The top row shows 18 CMCs in blue and 8 non-CMCs in red when Fadul 1-1 is treated as the reference and Fadul 1-2 the target. The bottom row shows the 17 CMCs and 13 non-CMCs when the roles are reversed. There is no discussion in Song (2013) about combining the results from these two comparison directions, but Tong, Song, and Chu (2015) propose using the minimum of the two CMC counts (17 in this example).

Similarly, CMCs and non-CMCs determined under the High CMC decision rule are shown in Figure 10. Treating Fadul 1-1 and Fadul 1-2 as the reference scan yields 20 and 18 CMCs, respectively. Combining the results as described above, the final High CMC count is 24.

In contrast, Figure 11 shows the CMC results for a comparison between Fadul 1-1 and a known non-match scan, Fadul 2-1, under the exact same processing conditions. Only two cells are classified as congruent matching cells under the original decision rule when Fadul 1-1 is the reference scan. No cells are classified as CMCs in the other direction. While not shown, this pair fails the High CMC criteria and thus was assigned 0 CMCs under the High CMC decision rule.

3 Discussion

Ambiguity in algorithmic descriptions

During the implementation process we encountered ambiguous descriptions of the various CMC pipelines. We include the pre-processing and cell-based comparison procedures in the description of CMC methodology to emphasize how sensitive the final results are to decisions made in these first two steps. The pre-processing and cell-based comparison procedures are discussed only briefly, if at all, in Song et al. (2014), Tong et al. (2014), Tong, Song, and Chu (2015), or Chen et al. (2017). However, the results reported often indicate a sensitivity to these procedures. Ambiguities range from minor implicit parameter choices (e.g., the convergence criteria for the robust Gaussian regression filter (Brinkman

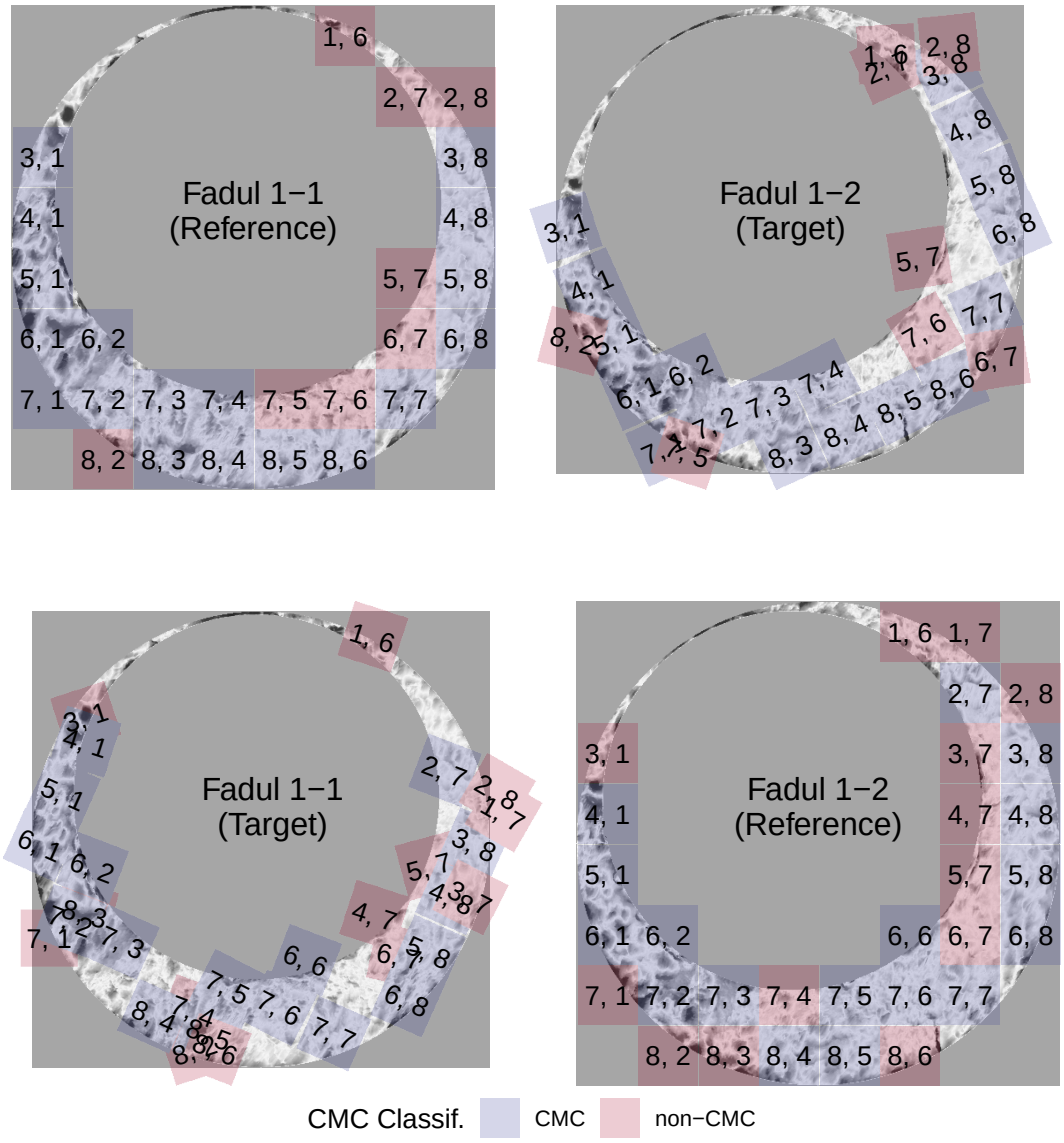


Figure 9: CMC results for the comparison between Fadul 1-1 and Fadul 1-2 using the original decision rule. The two plots in the top row show the 18 CMCs when Fadul 1-1 is treated as the "reference" cartridge case to which Fadul 1-2 (the "target") is compared. The second row shows the 17 CMCs when the roles are reversed. Red cells indicate where cells not identified as congruent achieve the maximum pairwise-complete correlation across all rotations of the target scan.

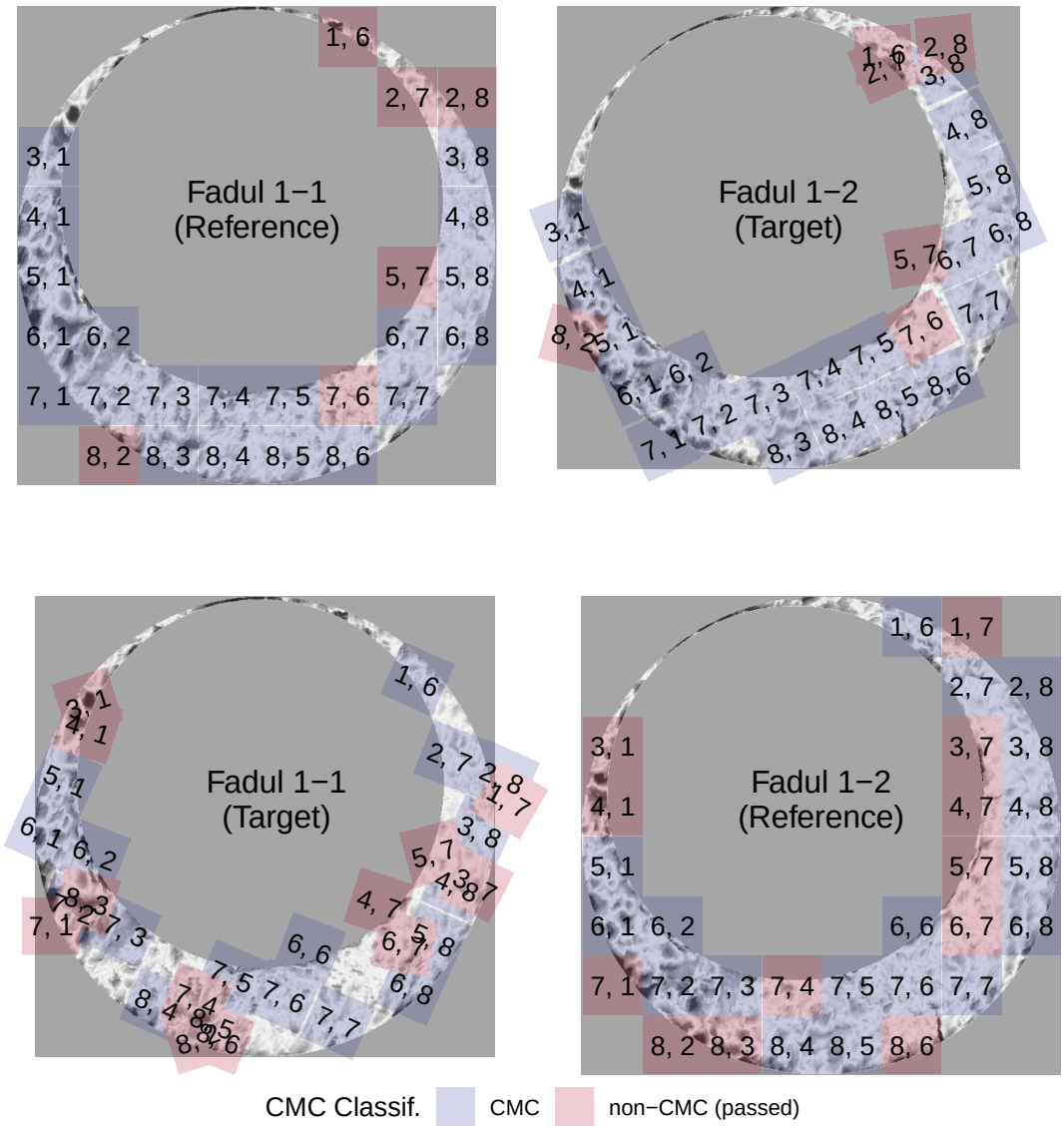


Figure 10: Applying the High CMC decision rule to the comparison of Fadul 1-1 and Fadul 1-2 results in 20 CMCs when Fadul 1-1 is treated as the reference (top) and 18 CMCs when Fadul 1-2 is treated as the reference (bottom). Although the individual comparisons do not yield considerably more CMCs than under the original CMC pipeline, Tong et al. (2015) indicate that the High CMCs from both comparisons are combined as the final High CMC count (each cell is counted at most once). Combining the results means that the High CMC decision rule tends to produce higher CMC counts than the original CMC pipeline. In this example, the combined High CMC count is 24 CMCs.

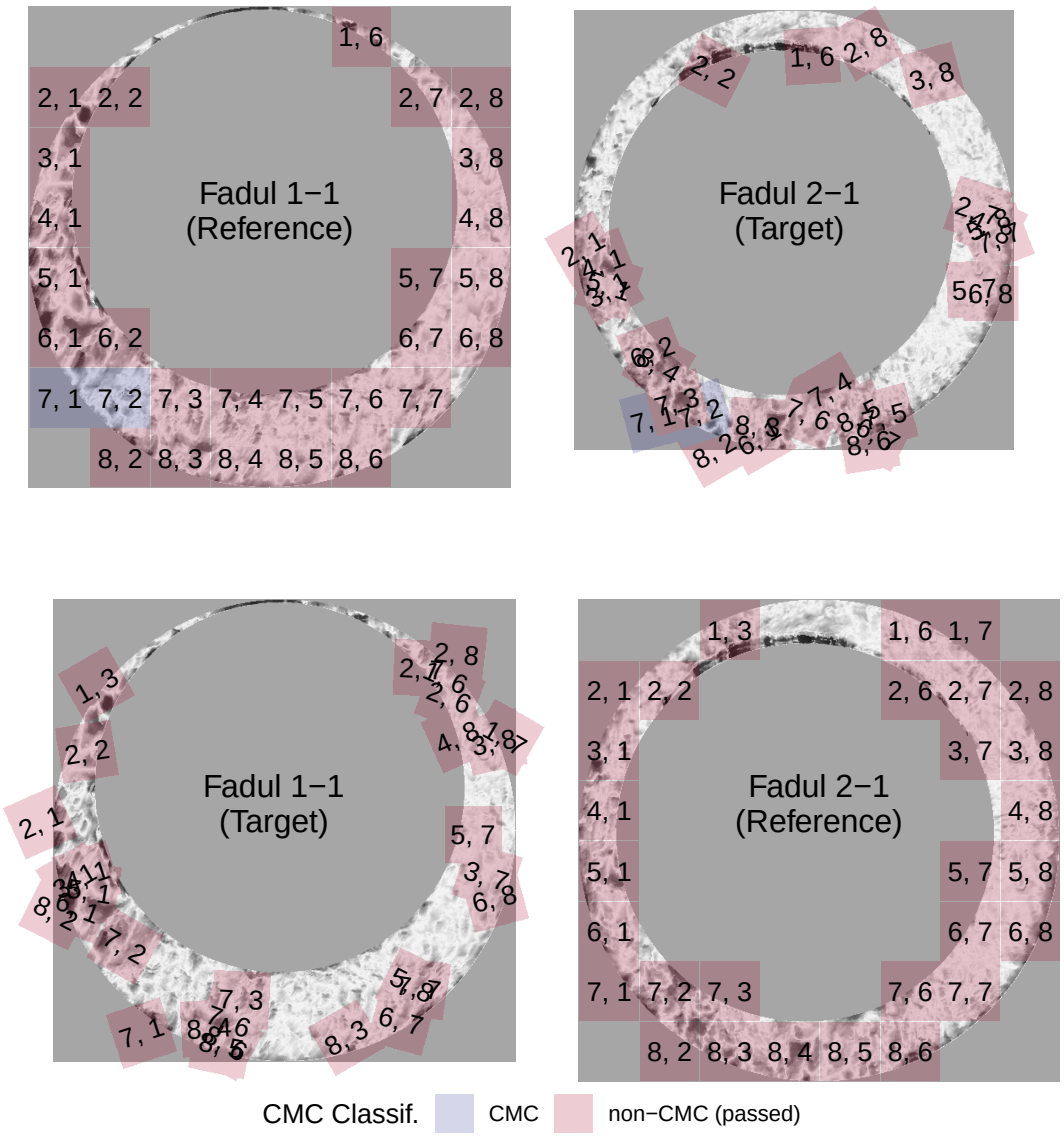


Figure 11: Applying both decision rules to the comparison between the non-match pair Fadul 1-1 and Fadul 2-1 results in 2 CMCs under the original decision rule (shown above) and 0 CMCs under the High CMC decision rule (not shown). The seemingly random behavior of the red cells exemplifies the assumption that cells in a non-match comparison do not exhibit an observable pattern. Random chance should be the prevailing factor in classifying non-match cells as CMCs.

and Bodschwinna 2003)) to procedures that are fundamental to feature calculation (e.g., how the cross-correlation is calculated). We bring up these ambiguities to demonstrate the difficulties that we faced when translating the conceptual description of the CMC pipeline into an actual pipeline. While many of these choices are unlikely to affect the results dramatically, we believe that any amount of variability that exists solely because of uncertainty in how the method was intended to be implemented is both unnecessary and dangerous in this application.

The only solution to such ambiguity is to enumerate, implement, and pare-down the possible choices that could have been made to arrive to published results. Unsurprisingly, this process takes a considerable amount of time and resources that would be better spent furthering the state of the field. During the creation of the **cmcR** package, the process of re-implementing the comparison and decision steps of the pipeline was fairly straightforward. Emulating the pre-processing procedures used, on the other hand, took months of trial and error. Even after this effort, we still have no assurances that our implementation would match the results of the original implementation if applied to other data sets.

In the next section, we describe the process of resolving these ambiguities in the CMC pipeline descriptions. In doing so, we abstract a set of principles by which pipelines and results can be rendered both computationally reproducible and more thoroughly understood.

CMC pattern matching pipeline

As described in the initial data section, the set of cartridge case scans from Fadul et al. (2011) is commonly used to compare the performance of various classification methods (Song et al. 2014; Tong, Song, and Chu 2015; Chen et al. 2017). This set consists of 40 cartridge cases and 780 total comparisons: 63 known match comparisons and 717 known non-match comparisons. Scans of each breech face impression were taken with a Nanofocus Confocal Light Microscope at 10 fold magnification for a nominal lateral resolution of 3.125 microns per pixel and published to the NBTRD (Zheng, Soons, and Thompson 2016). We also use the Weller et al. (2012) data set of 95 cartridge cases for comparison. For the Weller et al. (2012) dataset, we manually isolated the breech face impression regions using the FiX3P software (accessible here: `https://github.com/talenfisher/fix3p`). We compare results from the **cmcR** package to published results using processed scans available through the Iowa State University DataShare repository (Zemmels, Hofmann, and Vanderplas 2022). Our goal is to show that results obtained from **cmcR** are similar, at least qualitatively, to previously published results. However, justification for any differences will ultimately involve educated guesses due to the closed-source nature of the original implementations.

For each cartridge case pair, we calculate CMC counts under both the original and High CMC decision rules. In practice, we classify a cartridge case pair as "matching" if its CMC count surpasses some threshold; 6 CMCs being the generally accepted threshold in many papers (Tong, Song, and Chu 2015; Song et al. 2018; Song 2013). However, this threshold has been shown to not generalize well to all proposed methods and cartridge case data sets (Chen et al. 2017). We instead use an optimization criterion to select parameters. In doing so, we will demonstrate the sensitivity of the pipeline to parameter choice. Additionally, we introduce a set of principles designed to reduce the need for brute-force searches across parameter settings when re-implementing algorithms without accompanying code. Adherence to these principles yields not only computationally reproducible results, but also improves a reader's understanding of a proposed pipeline.

Processing condition sensitivity

Choosing threshold values $T_x, T_y, T_\theta, T_{\text{CCF}}$ for translation, rotation, and maximum cross-correlation is crucial in declaring a particular cell-region pair "congruent." However, many combinations of these thresholds yield perfect separation between the matching and non-matching CMC count distributions. Therefore, choosing parameters based on maximizing classification accuracy does not lead to an obvious, single set of parameters. We instead consider the ratio of between- and within-group variability to measure separation between match and non-match CMC counts.

Let C_{ij} denote the CMC count assigned to the jth cartridge case pair, $j = 1, ..., n_i$ from the ith group, $i = 1, 2$ representing matches and non-matches, respectively. For each set of thresholds we calculate the **Variance Ratio** r as:

$$r = r\left(T_x, T_y, T_\theta, T_{\text{CCF}}\right) = \frac{\sum_{i=1}^{2}\left(\overline{C}_{i.} - \overline{C}_{..}\right)^2}{\sum_{i=1}^{2}\frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(C_{ij} - \overline{C}_{i.}\right)^2},$$

where $\overline{C}_{i.}$ denotes the within-group CMC count average and $\overline{C}_{..}$ denotes the grand CMC count average. Greater separation between and less variability within the match and non-match CMC count

distributions yields larger r values.

For example, Figure 12 shows results for the original decision rule and the High CMC decision rule for parameters $T_x = 20 = T_y$ pixels, $T_{\text{CCF}} = 0.5$, and $T_\theta = 6$. Despite both decision rules resulting in separation between the matching and non-matching CMC count distributions, the High CMC decision rule yields greater separation as evidenced by the larger r value.

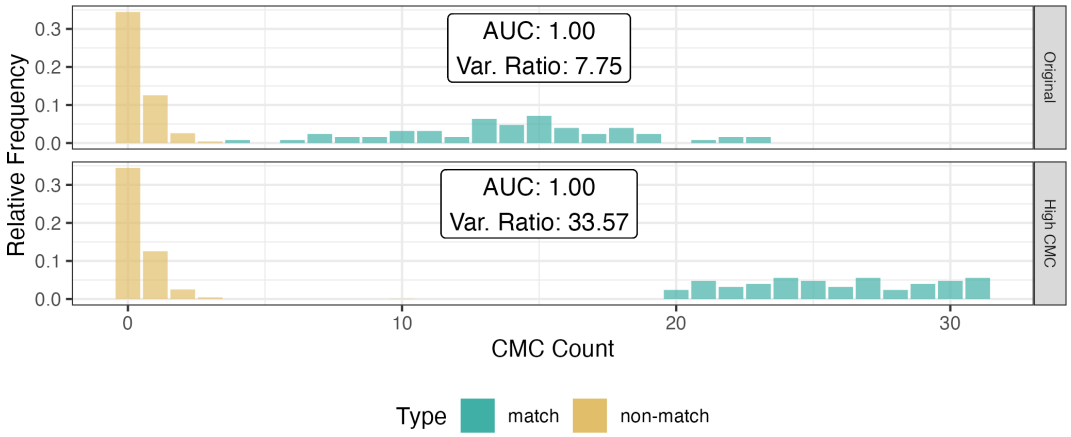


Figure 12: CMC count relative frequencies under the original decision rule and the High CMC decision rule for $T_{\Delta x} = 20 = T_{\Delta y}$ pixels, $T_{\text{CCF}} = 0.5$, and $T_\theta = 6$ degrees. An AUC $= 1$ corresponds to perfect separation of the match and non-match CMC count distributions. We can see that, for this set of processing parameters, the High CMC decision rule yields higher CMC counts for known matches that the original decision rule while known non-matches have the same distribution under both methods.

To explore the pipeline's sensitivity, we consider five dimensions that have a demonstrable impact on CMC counts:

- the decision rule (original or High CMC) used,
- whether the global trend is removed during pre-processing, and
- choice of congruency thresholds: translation T_x, T_y, rotation T_θ, and cross-correlation T_{CCF}.

Choosing a single parameter setting that results in perfect identification is not enough to generally understand the algorithm. Instead, we use the variance ratio r to identify promising ranges of parameters. Figure 13 shows the value of the variance ratio under different parameter settings. We see that the High CMC decision rule yields better separation than the original decision rule under any parameter setting. The largest variance ratio values are achieved for thresholds $T_x, T_y \in [10, 20]$, $T_\theta = 6$, and $T_{\text{CCF}} \in [0.4, 0.5]$. Interestingly, considering Table 3, only the parameters used in Song et al. (2018) fall into these ranges.

As shown in Figure 13, de-trending breech-scans in the pre-processing stage emerges as a critical step to achieve good algorithmic results. This step is not explicitly mentioned in the written-word descriptions of the algorithm in Song (2013), Tong et al. (2014), Tong, Song, and Chu (2015), Chen et al. (2017), or Song et al. (2018), though it appears from their examples that it was used in the process. Figure 13 also illustrates how breaking a pipeline up into modularized steps eases experimentation. We will expand upon this idea in the next section.

We compare the best results from **cmcR** to results presented in previous papers. In particular, we have calculated variance ratio statistics shown in Figure 14 based on CMC counts reported in Song (2013), Tong et al. (2014), Tong, Song, and Chu (2015), Chen et al. (2017), and Song et al. (2018). The last row in each facet shows the variance ratio values obtained from **cmcR**. We see that the implementation provided in **cmcR** yields comparable results to previous CMC papers.

4 Conclusion

Reproducibility is an indispensable component of scientific validity (Goodman, Fanelli, and Ioannidis 2016). In this paper, we demonstrate at least three ways reproducibility can go awry: ambiguity in procedural implementation, missing or incomplete data, and missing or incomplete code. In forensics, many matching algorithms are commonly presented in the form of conceptual descriptions with accompanying results. There is sound reasoning to this; conceptual descriptions are more easily understood by humans compared to computer code. However, using the CMC pipelines as an example

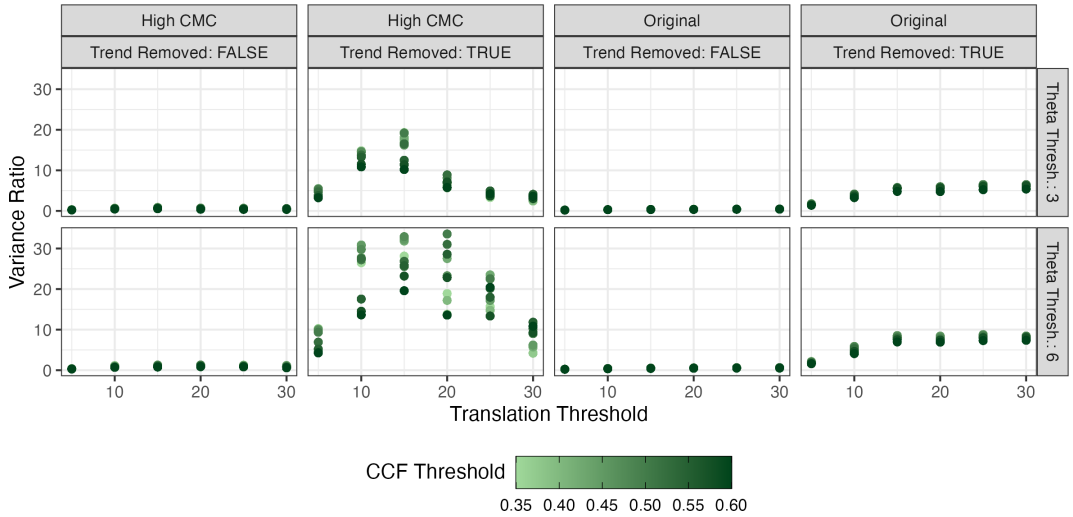


Figure 13: Variance ratio values are plotted for different parameter settings. High variance ratios are indicative of a a good separation between CMC counts for known matching pairs and known-non matching pairs. The High CMC decision rule generally performs better than the original decision rule. Removing the trend during pre-processing has a major impact on the effectiveness of the CMC pipeline. In this setting, translation thresholds $T_x, T_y \in [15, 20]$, a rotation threshold $T_\theta = 6$, and a CCF threshold $T_{\text{CCF}} \in [0.4, 0.5]$ lead to a separation of results.

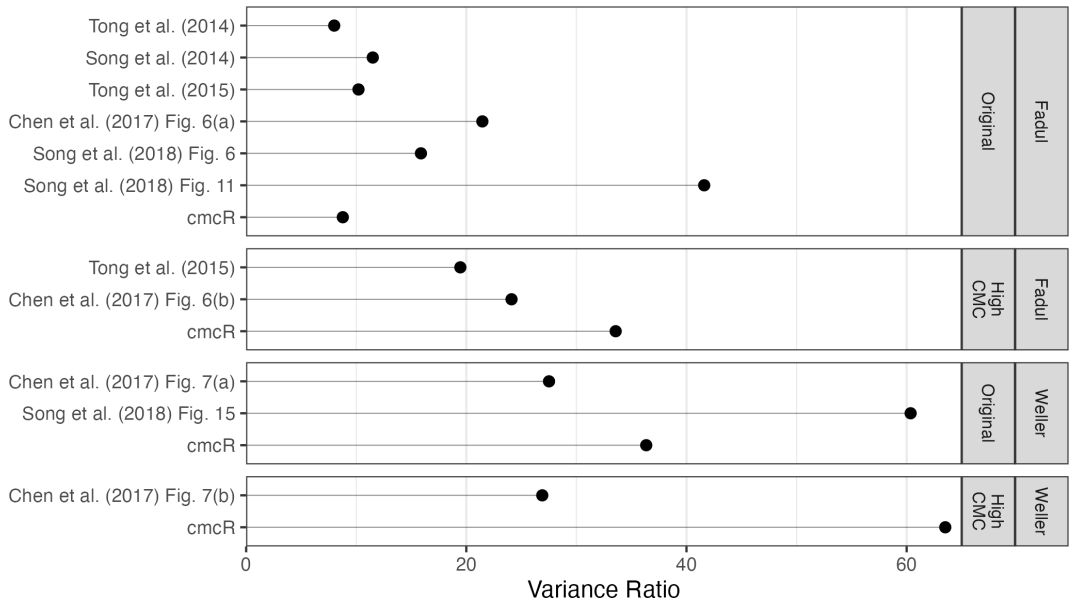


Figure 14: Variance ratios based on results reported in various CMC papers. The High CMC decision rule tends to outperform the original decision rule. However, it should be emphasized that each paper uses very different processing and parameter settings meaning the results are difficult to compare. The values labeled "cmcR" show the largest variance ratio values for the original and High CMC decision rules based on a limited grid search. These results indicate that the CMC pipeline implementation provided in **cmcR** yields comparable results to previous CMC papers.

we have observed the gaps that can exist when translating a conceptual description of an algorithm to a genuine implementation. This is largely due to the fact that conceptual descriptions rarely detail implicit parameter choices required to run an algorithm. Consequently, there are multiple choices that are compatible with the description of an algorithm in a publication. This is dangerous in a forensics context because if many parameter settings are valid but only a narrow range lead to the same conclusion, it is entirely possible that witnesses for the prosecution and defense comef to different conclusions. In order to prevent such misunderstandings, it is not enough to have guidelines for parameter settings and/or a sensitivity study – it is also necessary to standardize the specific computer code. The parameter values are only useful within the context of a single software package or pipeline.

These principles of open, accessible, interoperable code are also critical for a fair (in the legal sense) justice system: the defense has access to the code to understand the evidence against them, lawyers and examiners can assess the utility of the analysis method, and judges can determine whether a method is admissible in court. Transparent and intuitive open-source algorithms, such as **cmcR**, should be considered the gold standard in allowing the forensic science community to validate a pipeline.

Our contribution to the CMC literature is the open-source implementation, which fills the gaps in the human-friendly descriptions in the original papers. In addition, because we have structured the **cmcR** implementation as a modular pipeline, it is easier to improve upon the CMC method and document the effects of specific changes to the algorithm compared to previous versions. The modularization creates an explicit framework to assess the utility and effectiveness of each piece of the algorithm, and allows us to independently manipulate each step while monitoring the downstream impact on the results. Additionally, it allows future collaborators to improve on pieces of the pipeline, adding new options and improving the method without having to re-invent the wheel. Indeed, re-implementing steps of the pipeline is at best a useful academic exercise and at worst a waste of time and resources that could be spent actually improving the pipeline. Even after many months of trial and error, although we have succeeded in obtaining qualitatively similar results on two data sets, it is difficult to know whether our implementation will behave the same as previous implementations on external data sets. Generalizability is an important assessment for any computational algorithm (Vanderplas et al. 2020).

Our application is far from unique: some journals have adopted policies encouraging or requiring that authors provide code and data sufficient to reproduce the statistical analyses, with the goal of building a "culture of reproducibility" in their respective fields (Peng 2009, 2011; Stodden, Guo, and Ma 2013). Peer-review and scientific progress in the truest sense requires that *all* pre-processed data, code, and results be made openly available (Kwong 2017; Desai and Kroll 2017). Our experience with the CMC algorithm suggests that these standards should be adopted by the forensic science community, leveraging open-source ecosystems like R and software sharing platforms such as Github. We firmly believe that the forensic community should not go only halfway, trading a subjective, human black box for objective, proprietary algorithms that are similarly opaque and unauditable. Open, fully reproducible packages like **cmcR** allow research groups to make incremental changes, compare different approaches, and accelerate the pace of research and development.

5 Acknowledgement

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

We greatly appreciate the constructive feedback from the two anonymous reviewers. Special thanks also to all the developers and open-source contributors of R, knitr (Xie 2015, 2014), rticles (Allaire et al. 2021), and the tidyverse (Wickham et al. 2019), without whom this project would not have been possible.

6 Computational details

```
sessionInfo()

#> R version 4.2.2 (2022-10-31)
#> Platform: aarch64-apple-darwin20 (64-bit)
#> Running under: macOS Ventura 13.0
```

```
#>
#> Matrix products: default
#> BLAS:  /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] patchwork_1.1.2 rgl_1.0.1       x3ptools_0.0.3  forcats_1.0.0
#>  [5] stringr_1.5.0   dplyr_1.1.0     purrr_1.0.1     readr_2.1.3
#>  [9] tidyr_1.3.0     tibble_3.1.8    ggplot2_3.4.0   tidyverse_1.3.2
#> [13] cmcR_0.1.11
#>
#> loaded via a namespace (and not attached):
#>   [1] fs_1.6.1            lubridate_1.9.1     webshot_0.5.4
#>   [4] httr_1.4.4          hunspell_3.0.2      tools_4.2.2
#>   [7] backports_1.4.1     utf8_1.2.3          R6_2.5.1
#>  [10] DBI_1.1.3           colorspace_2.1-0    withr_2.5.0
#>  [13] readbitmap_0.1.5    gridExtra_2.3       tidyselect_1.2.0
#>  [16] compiler_4.2.2      extrafontdb_1.0     textshaping_0.3.6
#>  [19] quantreg_5.94       cli_3.6.0           rvest_1.0.3
#>  [22] SparseM_1.81        xml2_1.3.3          labeling_0.4.2
#>  [25] bookdown_0.32       scales_1.2.1        systemfonts_1.0.4
#>  [28] digest_0.6.31       tiff_0.1-11         yulab.utils_0.0.6
#>  [31] svglite_2.1.1       rmarkdown_2.20      base64enc_0.1-3
#>  [34] jpeg_0.1-10         pkgconfig_2.0.3     htmltools_0.5.4
#>  [37] extrafont_0.19      dbplyr_2.3.0        fastmap_1.1.0
#>  [40] htmlwidgets_1.6.1   rlang_1.0.6         readxl_1.4.2
#>  [43] rstudioapi_0.14     farver_2.1.1        gridGraphics_0.5-1
#>  [46] generics_0.1.3      zoo_1.8-11          jsonlite_1.8.4
#>  [49] googlesheets4_1.0.1 magrittr_2.0.3      kableExtra_1.3.4
#>  [52] ggplotify_0.1.0     Matrix_1.5-3        Rcpp_1.0.10
#>  [55] munsell_0.5.0       fansi_1.0.4         ggnewscale_0.4.8
#>  [58] lifecycle_1.0.3     stringi_1.7.12      yaml_2.3.7
#>  [61] MASS_7.3-58.2       grid_4.2.2          crayon_1.5.2
#>  [64] lattice_0.20-45     cowplot_1.1.1       splines_4.2.2
#>  [67] haven_2.5.1         hms_1.1.2           magick_2.7.3
#>  [70] knitr_1.42          pillar_1.8.1        igraph_1.3.5
#>  [73] codetools_0.2-19    imager_0.42.18      reprex_2.0.2
#>  [76] glue_1.6.2          evaluate_0.20       rjtools_1.0.9.9001
#>  [79] bmp_0.3             BiocManager_1.30.19 modelr_0.1.10
#>  [82] vctrs_0.5.2         png_0.1-8           tzdb_0.3.0
#>  [85] yesno_0.1.2         MatrixModels_0.5-1  Rttf2pt1_1.3.12
#>  [88] cellranger_1.1.0    gtable_0.3.1        assertthat_0.2.1
#>  [91] xfun_0.37           cranlogs_2.1.1      broom_1.0.3
#>  [94] pracma_2.4.2        ragg_1.2.5          viridisLite_0.4.1
#>  [97] survival_3.5-0      googledrive_2.0.0   gargle_1.3.0
#> [100] timechange_0.2.0    ellipsis_0.3.2
```

References

Allaire, JJ, Yihui Xie, R Foundation, Hadley Wickham, Journal of Statistical Software, Ramnath Vaidyanathan, Association for Computing Machinery, et al. 2021. *Rticles: Article Formats for r Markdown*. https://CRAN.R-project.org/package=rticles.

Barthelme, Simon. 2019. *Imager: Image Processing Library Based on 'CImg'*. https://CRAN.R-project.org/package=imager.

Brigham, E. Oran. 1988. *The Fast Fourier Transform and Its Applications*. USA: Prentice-Hall, Inc.

Brinkman, S., and H. Bodschwinna. 2003. "Advanced Gaussian Filters." In *Advanced Techniques for Assessment Surface Topography: Development of a Basis for 3D Surface Texture Standards "SURFSTAND"*,

edited by L. Blunt and X. Jiang. United States: Elsevier Inc. https://doi.org/10.1016/B978-1-903996-11-9.X5000-2.

Brown, Lisa Gottesfeld. 1992. "A Survey of Image Registration Techniques." *ACM Computing Surveys* 24: 325–76. https://dl.acm.org/doi/10.1145/146370.146374.

Chen, Zhe, John Song, Wei Chu, Johannes A. Soons, and Xuezeng Zhao. 2017. "A Convergence Algorithm for Correlation of Breech Face Images Based on the Congruent Matching Cells (CMC) Method." *Forensic Science International* 280 (November): 213–23. https://doi.org/10.1016/j.forsciint.2017.08.033.

Chu, Wei, Mingsi Tong, and John Song. 2013. "Validation Tests for the Congruent Matching Cells (CMC) Method Using Cartridge Cases Fired with Consecutively Manufactured Pistol Slides." *Journal of the Association of Firearms and Toolmarks Examiners* 45 (4): 6. https://www.nist.gov/publications/validation-tests-congruent-matching-cells-cmc-method-using-cartridge-cases-fired.

Crawford, Amy. 2020. "Bayesian Hierarchical Modeling for the Forensic Evaluation of Handwritten Documents." {Ph.D thesis}, Iowa State University. https://doi.org/10.31274/etd-20200624-257.

Curran, James Michael, Tacha Natalie Hicks Champod, and John S Buckleton, eds. 2000. *Forensic Interpretation of Glass Evidence*. Boca Raton, FL: CRC Press.

Desai, Deven R., and Joshua A. Kroll. 2017. "Trust but Verify: A Guide to Algorithms and the Law." *Harvard Journal of Law & Technology (Harvard JOLT)* 31 (1): 1–64.https://ssrn.com/abstract=2959472.

Fadul, T., G. Hernandez, S. Stoiloff, and Gulati Sneh. 2011. "An Empirical Study to Improve the Scientific Foundation of Forensic Firearm and Tool Mark Identification Utilizing 10 Consecutively Manufactured Slides." https://www.ojp.gov/ncjrs/virtual-library/abstracts/empirical-study-improve-scientific-foundation-forensic-firearm-and.

"Geometrical product specifications (GPS) — Surface texture: Areal — Part 72: XML file format x3p." 2017. Standard. Vol. 2014. Geneva, CH: International Organization for Standardization. https://www.iso.org/standard/62310.html.

Goodman, Steven N., Daniele Fanelli, and John P. A. Ioannidis. 2016. "What Does Research Reproducibility Mean?" *Science Translational Medicine* 8 (341): 341ps12–12. https://doi.org/10.1126/scitranslmed.aaf5027.

Goor, Robert, Douglas Hoffman, and George Riley. 2020. "Novel Method for Accurately Assessing Pull-up Artifacts in STR Analysis." *Forensic Science International: Genetics* 51 (November): 102410. https://doi.org/10.1016/j.fsigen.2020.102410.

Hare, Eric, Heike Hofmann, and Alicia Carriquiry. 2017. "Automatic Matching of Bullet Land Impressions." *The Annals of Applied Statistics* 11 (4): 2332–56. http://arxiv.org/abs/1601.05788.

Hesselink, Wim H., Arnold Meijster, and Coenraad Bron. 2001. "Concurrent Determination of Connected Components." *Science of Computer Programming* 41 (2): 173–94. https://doi.org/10.1016/S0167-6423(01)00007-7.

Hofmann, Heike, Susan Vanderplas, Ganesh Krishnan, and Eric Hare. 2020. *x3ptools: Tools for Working with 3D Surface Measurements*. https://cran.r-project.org/web/packages/x3ptools/index.html.

Kwong, Katherine. 2017. "The Algorithm Says You Did It: The Use of Black Box Algorithms to Analyze Complex DNA Evidence Notes." *Harvard Journal of Law & Technology (Harvard JOLT)* 31 (1): 275–302. https://jolt.law.harvard.edu/assets/articlePDFs/v31/31HarvJLTech275.pdf.

Leek, Jeffrey T., and Leah R. Jager. 2017. "Is Most Published Research Really False?" *Annual Review of Statistics and Its Application* 4 (1): 109–22. https://doi.org/10.1146/annurev-statistics-060116-054104.

National Academy of Sciences, Engineering, and Medicine. 2019. *Reproducibility and Replicability in Science*. National Academies Press. https://doi.org/10.17226/25303.

National Research Council. 2009. *Strengthening Forensic Science in the United States: A Path Forward*. Washington, DC: The National Academies Press. https://doi.org/10.17226/12589.

Ott, Daniel, Robert Thompson, and Junfeng Song. 2017. "Applying 3D Measurements and Computer Matching Algorithms to Two Firearm Examination Proficiency Tests." *Forensic Science International* 271 (February): 98–106. https://doi.org/10.1016/j.forsciint.2016.12.014.

Park, Soyoung, and Alicia Carriquiry. 2020. "An Algorithm to Compare Two-Dimensional Footwear Outsole Images Using Maximum Cliques and Speeded-up Robust Feature." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 13 (2): 188–99. https://doi.org/10.1002/sam.11449.

Park, Soyoung, and Sam Tyner. 2019. "Evaluation and Comparison of Methods for Forensic Glass Source Conclusions." *Forensic Science International* 305 (December): 110003. https://doi.org/10.1016/j.forsciint.2019.110003.

Peng, Roger D. 2009. "Reproducible Research and Biostatistics." *Biostatistics* 10 (3): 405–8. https://doi.org/10.1093/biostatistics/kxp014.

———. 2011. "Reproducible Research in Computational Science." *Science* 334 (6060): 1226–27. https://www.jstor.org/stable/41352177.

President's Council of Advisors on Sci. & Tech. 2016. "Forensic Science in Criminal Courts: Ensuring Scientific Validity of Feature-Comparison Methods." https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/PCAST/pcast_forensic_science_report_final.pdf.

Song, John. 2013. "Proposed 'NIST Ballistics Identification System (NBIS)' Based on 3D Topography Measurements on Correlation Cells." *American Firearm and Tool Mark Examiners Journal* 45 (2): 11. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=910868.

Song, John, Wei Chu, Mingsi Tong, and Johannes Soons. 2014. "3D Topography Measurements on Correlation Cells—a New Approach to Forensic Ballistics Identifications." *Measurement Science and Technology* 25 (6): 064005. https://doi.org/10.1088/0957-0233/25/6/064005.

Song, John, Theodore V. Vorburger, Wei Chu, James Yen, Johannes A. Soons, Daniel B. Ott, and Nien Fan Zhang. 2018. "Estimating Error Rates for Firearm Evidence Identifications in Forensic Science." *Forensic Science International* 284 (March): 15–32. https://doi.org/10.1016/j.forsciint.2017.12.013.

Stodden, Victoria, Peixuan Guo, and Zhaokun Ma. 2013. "Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals." Edited by Dmitri Zaykin. *PLoS ONE* 8 (6): e67111. https://doi.org/10.1371/journal.pone.0067111.

Tai, Xiao Hui. 2021. *cartridges3D: Algorithm to Compare Cartridge Case Images*. https://github.com/xhtai/cartridges3D.

Tai, Xiao Hui, and William F. Eddy. 2018. "A Fully Automatic Method for Comparing Cartridge Case Images," *Journal of Forensic Sciences* 63 (2): 440–48. http://doi.wiley.com/10.1111/1556-4029.13577.

The Linux Foundation. 2017. "Using open source software to speed up development and gain business advantage." https://www.linuxfoundation.org/blog/using-open-source-software-to-speed-development-and-gain-business-advantage/.

Thompson, Robert. 2017. *Firearm Identification in the Forensic Science Laboratory*. National District Attorneys Association. https://doi.org/10.13140/RG.2.2.16250.59846.

Tong, Mingsi, John Song, and Wei Chu. 2015. "An Improved Algorithm of Congruent Matching Cells (CMC) Method for Firearm Evidence Identifications." *Journal of Research of the National Institute of Standards and Technology* 120 (April): 102. https://doi.org/10.6028/jres.120.008.

Tong, Mingsi, John Song, Wei Chu, and Robert M. Thompson. 2014. "Fired Cartridge Case Identification Using Optical Images and the Congruent Matching Cells (CMC) Method." *Journal of Research of the National Institute of Standards and Technology* 119 (November): 575. https://doi.org/10.6028/jres.119.023.

Tvedebrink, Torben, Mikkel Meyer Andersen, and James Michael Curran. 2020. "DNAtools: Tools for Analysing Forensic Genetic DNA Data." *Journal of Open Source Software* 5 (45): 1981. https://doi.org/10.21105/joss.01981.

Tyner, Sam, Soyoung Park, Ganesh Krishnan, Karen Pan, Eric Hare, Amanda Luby, Xiao Hui Tai, Heike Hofmann, and Guillermo Basulto-Elias. 2019. "Sctyner/OpenForSciR: Create DOI for Open Forensic Science in r." Zenodo. https://doi.org/10.5281/ZENODO.3418141.

Vanderplas, Susan, Melissa Nally, Tylor Klep, Cristina Cadevall, and Heike Hofmann. 2020. "Comparison of Three Similarity Scores for Bullet LEA Matching." *Forensic Science International*, January. https://doi.org/10.1016/j.forsciint.2020.110167.

Weller, Todd J., Alan Zheng, Robert Thompson, and Fred Tulleners. 2012. "Confocal Microscopy Analysis of Breech Face Marks on Fired Cartridge Cases from 10 Consecutively Manufactured Pistol Slides" 57 (4). https://doi.org/10.1111/j.1556-4029.2012.02072.x.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

Zemmels, Joseph, Heike Hofmann, and Susan Vanderplas. 2022. "Zemmels et al. (2022) Cartridge Case Scans." Iowa State University. https://doi.org/10.25380/IASTATE.19686297.V1.

Zheng, Xiaoyu A., Johannes A. Soons, and Robert M. Thompson. 2016. "NIST Ballistics Toolmark Research Database." https://tsapps.nist.gov/NRBTD/.

Joseph Zemmels
Center for Statistics and Applications in Forensic Evidence
Department of Statistics
Iowa State University
2438 Osborn Drive
Ames, IA 50011

jzemmels@iastate.edu

Susan VanderPlas
University of Nebraska - Lincoln
Department of Statistics
340 Hardin Hall North Wing
3310 Holdrege St.
Lincoln, NE 68583
susan.vanderplas@unl.edu

Heike Hofmann
Center for Statistics and Applications in Forensic Evidence
Department of Statistics
Iowa State University
2438 Osborn Drive
Ames, IA 50011
hofmann@iastate.edu