QC-MDPC codes DFR and the IND-CCA security of BIKE

Valentin Vasseur

October 29, 2021

Preamble

The aim of this document is to clarify the DFR (Decoding Failure Rate) claims made for BIKE, a third round alternate candidate KEM (Key Encapsulation Mechanism) to the NIST call for post-quantum cryptography standardization. For the most part, the material presented here is not new, it is extracted from the relevant scientific literature, in particular [Vas21].

Even though a negligible DFR is not needed for a KEM using ephemeral keys (*e.g.* TLS) which only requires IND-CPA security, it seems that IND-CCA security, relevant for reusable/static keys, has become a requirement. Therefore, a negligible DFR is needed both for the security reduction [FO99; HHK17] and to thwart existing attacks [GJS16].

Proving a DFR lower than $2^{-\lambda}$ where λ is the security parameter (*e.g.* $\lambda = 128$ or 256) is hardly possible with mere simulation. Instead a methodology based on modelization, simulation, and extrapolation with confidence estimate was devised [Vas21]. Models are backed up by theoretical results [Til18; SV19], but do not account for some combinatorial properties of the underlying error correcting code. Those combinatorial properties give rise to what is known in telecommunication as “error floors” [Ric03].

The statistical modeling predicts a fast decrease of the DFR as the block size grows, the *waterfall* region, whereas the combinatorial properties, weak keys[DGK19] or near-codewords[Vas21], predict a slower decrease, the *error floor* region. The issue here is to show that the error floor occurs in a region where the DFR is already below the security requirement. This would validate the extrapolation approach, and, as far as we can say, this appears to be the case for the QC-MDPC codes corresponding to BIKE parameters.

The impact of the QC-MDPC code combinatorial properties on decoding, as reported in this document, is better and better understood. In particular, it strongly relates with the spectrum of low weight vectors, as defined in [GJS16]. At this point, none of the results we are aware of and which are presented here contradict in any way the DFR claims made for BIKE. Admittedly those claims remain heuristic in part, but could be understood as an additional assumption, just like the computational assumptions made for all similar primitives, under which the BIKE scheme is IND-CCA secure.

1 Introduction

BIKE is a code-based key encapsulation mechanism based on Quasi-Cyclic Moderate Density Parity-Check (QC-MDPC) codes [MTSB13] submitted to the NIST standardization process on post-quantum cryptography [BIKE]. These are related to Low Density Parity Check (LDPC) codes, but instead of admitting a sparse parity check matrix (with rows of constant small weight), they admit a somewhat sparse parity check matrix, typically with rows of Hamming weight $O(\sqrt{n})$ with respect to a code length of n. Together with a quasi-cyclic structure, they allow the design of a McEliece-like public-key encryption scheme [McE78] with reasonable key size.

Security of a cryptosystem is usually considered by the hardness of solving a problem and the number of bits of security it provides is estimated with the complexity of solving it with the best known algorithms. In the case of a cryptosystem with a non-zero probability of decryption failure, higher security requirements demand also that this probability of failure be negligible.

BIKE is parameterized mainly by three values: t the Hamming weight of the error vector, w the row weight of the parity check matrix, and r the block length. To achieve IND-CPA security, the parameter t and w should be chosen to match a certain security level λ. To have IND-CCA security, we also need to make sure the decryption failure probability is less than $2^{-\lambda}$. We have $t \approx w$ and they are roughly equal to λ. Increasing the block length r has only a small impact on the computational hardness of the underlying problems, but it can significantly reduce the decryption failure probability. Thus, in this document, we consider that t and w are fixed and that only r varies.

Decryption in BIKE is done by decoding a QC-MDPC code which is usually done using variants of the bit-flipping algorithm (see Algorithm 1, or [DGK20; BIKE] for the complete description of the BGF variant selected for the NIST submission). Proving a low Decoding Failure Rate (DFR) is challenging, the decoders that can be analyzed often have rather poor performance, either because they are sequential or because they are limited to a small number of iterations. The better performance achieved by the latest developments in decoding algorithms has reduced the parameters size, but they are quite less amenable to analysis.

The typical DFR curve resembles the one shown in the Figure 1. It is strictly decreasing but goes through two different regimes: first it is concave with a very steep slope (the waterfall) then it has an inflection point and the slope becomes less steep (the error floor). As long as the target DFR is still in the waterfall regime, it is possible to use the simulation data (the DFR at block lengths r and r') to obtain the block length r_{ext} where the DFR is $2^{p_{\text{ext}}} < 2^{-\lambda}$. Indeed, as the curve is concave, the line passing through these two points is above it. We can get closer to the optimal block length r_λ by taking larger values for r and r'. However, this is limited by the computing power that one can devote to simulations.

As it is not feasible to exhaustively cover all possible QC-MDPC codes and error patterns, the DFRs at block lengths r and r' are obtained by drawing a large sample uniformly at random. Both the measures and the extrapolation thus have confidence intervals, the derivation of which will be discussed in §3.

We reduce the extrapolation approach to a simple assumption of concavity of the DFR curve. This assumption is supported by analytical results. We study

Algorithm 1: Parallel bit-flipping algorithm.

```
function parallel_bitflip($\mathbf{H}, \mathrm{s}$):
    input  : A sparse parity check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$,
             a syndrome $\mathrm{s} = \mathbf{H}\mathrm{e}^\top \in \mathbb{F}_2^{n-k}$.
    output : An error pattern $\mathrm{e}' \in \mathbb{F}_2^n$ such that $\mathbf{H}\mathrm{e}'^\top = \mathrm{s}$.
    $\mathrm{e}' \leftarrow 0$;
    $\mathrm{s}' \leftarrow \mathrm{s}$;
    $T \leftarrow$ threshold($context$);
    while $\mathrm{s}' \neq 0$ do
        for $j \in \{0, \dots, n-1\}$ do
            if $|\mathbf{h}_j \star \mathrm{s}'| \geq T$ then
                $e'_j \leftarrow 1 - e'_j$ ;
        $\mathrm{s}' \leftarrow \mathrm{s} - \mathbf{H}\mathrm{e}'^\top$;
    return $\mathrm{e}'$;
```

two phenomena that are most susceptible to challenge this assumption.

The first one is weak keys, which occurs because a certain structure in the parity check matrix of the QC-MDPC code results in a higher than average DFR. We show, in §A, some new properties on the distance spectrum originally defined in [GJS16]. The definition of the weak keys and the assessment of their impact on their impact on the DFR will be discussed in §4.

The second phenomenon we discuss is the error floor already mentioned with Figure 1. Indeed, if the target DFR is not in the waterfall region, we cannot use the extrapolation method. This error floor is a familiar occurrence for LDPC codes and it may be attributed to the presence of low weight codewords or low weight error patterns yielding a low weight syndrome (the near-codewords). We show that the former are harmless and identify patterns that correspond to the latter. Again, we evaluate their impact on the DFR, in §5.

Previous works have been conducted on the weak keys of QC-MDPC schemes. First, we can mention two classes of weak keys that are algebraic in nature. Their structure allows an attacker to fully recover the private key in polynomial time as long as the private key $(\mathrm{h}_0, \mathrm{h}_1) \in \mathbb{F}_2[x]/(x^r - 1)$ has a representation[1] such that

$$\deg(h_0) + \deg(h_1) < r \quad \text{or} \quad \deg(h_1) + \deg(h_0^{-1}) < r\,.$$

respectively in the case of [BDLO16] and [AYU20]. In the first case, the density of weak keys and the cost of recovery algorithms are such that an attack would have an average time complexity beyond 2^λ (for λ the security parameter). In the second case, the density of the keys is so negligible that there can be no realistic concern. Other weak keys have been defined in [DGK19], they are of a different nature. Their characteristic is that they produce a code with a higher than average failure rate, which can be exploited with a decryption oracle in an attack à la [GJS16]. The keys presented in this document are a generalization of

[1] Equivalent representations of keys are obtained by shifting or multiplying the monomial exponent of the polynomials by the same constants, see [BDLO16] or §4.

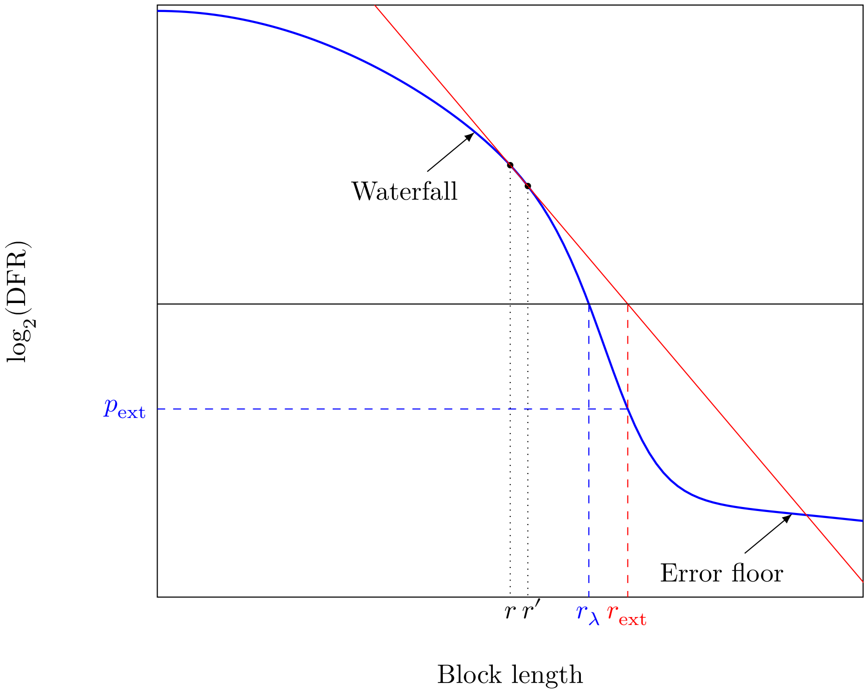

Figure 1: Typical DFR curve.

those and have been the subject of a preprint [SV20]. Finally, a successful attack involving weak keys has been conducted on LEDAcrypt [APRS20], a scheme that uses QC-LDPC codes that are related to QC-MDPC codes. The problem lies in the additional structure of the code, a characteristic that is not present in QC-MDPC codes.

This paper reports the results previously published in [Vas21] in a more synthetic and self-contained version. In addition, we provide data on the Level 3 and 5 parameter sets rather than just Level 1. The figures in this document compile extensive simulation data that can be replicated using the full C source code available at https://github.com/vvasseur/qcmdpc_decoder.

Outline.

This article starts with some reminders on the security of BIKE in §2. We justify the methodology of fixing all parameters but the block size r by giving a quick summary of the underlying problems, the best known attacks, and the security reduction of the $\mathsf{FO}^{\not\perp}$ transformation. We also give our definition of a weak key in the decoding context and we characterize the problematic error patterns in relation to the error floor. We show the link between this spectrum and the number of intersections between two columns of a circulant matrix, as well as a way to accurately count specific patterns that is needed in §4 & §5. In §3, we develop our DFR extrapolation method that is built on a decoding assumption and detail how to derive confidence intervals on it. Weak keys are constructed and evaluated in §4 for the BGF decoder with BIKE Level 1-parameters. Finally, in §5, we show that the quasi-cyclic structure of the code implies the existence of near-codewords. As with weak keys, we construct and evaluate specific error patterns, likely to trigger an error floor, built from these near-codewords. In appendix, we prove, in §A, new properties on the distance spectrum previously defined in [GJS16]. All simulation results are gathered in §B.

Notation.

For any binary vector v, we denote v_i its i-th coordinate and $|\mathrm{v}|$ its Hamming weight. Moreover, we will identify v with its support, that is $i \in \mathrm{v}$ if and only if $v_i = 1$. Given two binary vectors u and v of same length, we will denote $\mathrm{u} \star \mathrm{v}$ their component-wise product as vectors or, equivalently for binary vectors, the set of all indices that belong to both u and v.

We also adopt the following notations in this document.

- The decoding failure rate for a QC-MDPC, with a set of keys $\mathcal{H}$ and a set of messages $\mathcal{E}$ for a decoder $\mathcal{D}$ is written as

$$\mathrm{DFR}_{\mathcal{D},\mathcal{H}}^{\mathcal{E}} := \Pr\left[\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^\top) \neq \mathrm{e} \,\middle|\, \mathbf{H} \in \mathcal{H}, \mathrm{e} \in \mathcal{E}\right] .$$

- For ease of reading, when omitted, the set $\mathcal{E}$ is by default the whole set of admissible messages

$$\mathcal{E} = \mathcal{E}_{n,t} := \{\mathrm{e} \in \{0,1\}^n \mid |\mathrm{e}| = t\} .$$

- When omitted, the set $\mathcal{H}$ is by default the whole set of admissible keys

$$\begin{aligned}&\mathcal{H} = \mathcal{H}_{d,w,r\times n}\\ &:= \left\{\mathbf{H} \in \mathbb{F}_2^{r\times n} \,\middle|\, \forall i \in \{0,\dots,r-1\}, \left|\mathbf{h}_i^\top\right| = w, \forall j \in \{0,\dots,n-1\}, \left|\mathbf{h}_j\right| = d\right\} .\end{aligned}$$

- When the column weight d, the row weight w, and the error weight t are obvious from the context, we will write

$$\mathrm{DFR}(r) := \mathrm{DFR}_{\mathcal{D},\mathcal{H}_{d,2d,r\times 2r}}^{\mathcal{E}_{2r,t}} .$$

2 Reminders on the security of BIKE

The security of McEliece or Niederreiter type cryptosystems relates to the generic problems of decoding linear codes and finding low weight codewords. They were proven to be NP-complete in [BMT78]. For BIKE, we have to consider the quasi-cyclic versions[2] of these two problems $\mathrm{SD}(n,k,t)$ and $\mathrm{CF}(n,k,w)$.

Fujisaki and Okamoto proposed, in [FO99], an IND-CCA secure hybrid encryption scheme by combining a one-way secure scheme and a symmetric encryption primitive. This was later restated in a more modern way by Dent in [Den03]. Finally, a "modular" analysis was presented in [HHK17], it provides tighter security reductions and takes into account decryption errors (through a notion called δ-correctness).

[2] The elements of $\mathbb{F}_2[x]/(x^r-1)$ of odd and even weight are respectively denoted $\mathbb{F}_2[x]/(x^r-1)_{\mathrm{odd}}$ and $\mathbb{F}_2[x]/(x^r-1)_{\mathrm{even}}$. For any integer t, its parity is denoted $\mathrm{p}(t) \in \{\mathrm{odd}, \mathrm{even}\}$

Problem 1. *Syndrome Decoding* – $\mathrm{SD}(n, k, t)$
Instance: *A parity check matrix* $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$*, a syndrome* $\mathrm{s} \in \mathbb{F}_2^{n-k}$*, a target weight* $t > 0$.
Property: *There exists* $\mathrm{e} \in \mathbb{F}_2^n$ *such that* $|\mathrm{e}| = t$ *and* $\mathbf{H}\mathrm{e}^\top = \mathrm{s}$.

Problem 2. *Codeword Finding* – $\mathrm{CF}(n, k, w)$
Instance: *A parity check matrix* $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$*, a target weight* $w > 0$.
Property: *There exists* $\mathrm{e} \in \mathbb{F}_2^n$ *such that* $|\mathrm{e}| = w$ *and* $\mathbf{H}\mathrm{e}^\top = 0$.

Figure 2: Hard problems in code-based cryptography.

Problem 3. $(2,1)$-*QC Syndrome Decoding* – $(2,1)$-*QCSD*
Instance: $(h, s) \in \mathbb{F}_2[x]/(x^r-1)_{\mathrm{odd}} \times \mathbb{F}_2[x]/(x^r-1)_{\mathrm{p}(t)}$*, an integer* $t > 0$*.*
Property: *There exists* $(e_0, e_1) \in \mathrm{E}_t$ *such that* $e_0 + e_1 h = s$*.*

Problem 4. $(2,1)$-*QC Codeword Finding* – $(2,1)$-*QCCF*
Instance: $h \in \mathbb{F}_2[x]/(x^r-1)_{\mathrm{odd}}$*, an even integer* $w > 0$*, with* $d := w/2$ *odd.*
Property: *There exists* $(h_0, h_1) \in \mathcal{H}_{d,w}$ *such that* $h_1 + h_0 h = 0$*.*

Figure 3: Hard problems in code-based cryptography.

Definition 1. A KEM $(\mathrm{KeyGen}, \mathrm{Encaps}, \mathrm{Decaps})$ is said to be δ-correct if

$$\Pr\left[\mathrm{Decaps}(\mathtt{sk}, c) \neq K \,\middle|\, (\mathtt{pk}, \mathtt{sk}) \leftarrow \mathrm{KeyGen}(\{0,1\}^\lambda); (K, c) \leftarrow \mathrm{Encaps}(\mathtt{pk})\right] \leq \delta\,.$$

For BIKE, as the encapsulation includes a randomization of the error pattern, if the decoder used has an average DFR (for any parity check matrix and any error pattern) of δ then the KEM is δ-correct.

BIKE uses the $\mathsf{FO}^{\not\perp}$ transformation, a way to transform a PKE into a KEM with implicit rejection, using two hash functions H, K. If the original PKE is IND-CPA secure and δ-correct, it is shown in [HHK17] that the resulting KEM is IND-CCA secure under certain conditions fairly easily obtained. The security of the system can be summarized as the following theorem.

Theorem 2 (Theorem 3.2 & Theorem 3.4 in [HHK17] & §C.1.2 in [BIKE])**.** *If* BIKE *is* δ*-correct, then for any IND-CCA adversary* $\mathcal{B}$ *against it, issuing at most* q_{K} *and respectively* q_{H} *queries to the random oracle* K *and respectively* H *there exists an IND-CPA adversary* $\mathcal{A}$ *against* $\mathrm{QCSD}_{r,t}$ *and a distinghuer* $\mathcal{D}$ *against* $\mathrm{QCCF}_{r,w}$ *running in about the same time as* $\mathcal{B}$ *such that*

$$\mathrm{Adv}_{\mathrm{KEM}}^{\mathrm{IND-CCA}}(\mathcal{B}) \leq q_{\mathrm{H}} \cdot \delta + \frac{2q_{\mathrm{H}} + q_{\mathrm{K}} + 1}{|\mathcal{M}|} + \frac{3}{2}\mathrm{Adv}_{\mathrm{QCSD}_{r,t}}^{\mathrm{OW}}(\mathcal{A}) + \frac{3}{2}\mathrm{Adv}_{\mathrm{QCCF}_{r,w}}^{\mathrm{IND}}(\mathcal{D})\,.$$

Here $\mathrm{Adv}_{\mathrm{QCSD}_{r,t}}^{\mathrm{OW}}(\mathcal{A})$ relates to the advantage of adversary $\mathcal{A}$ challenging the one-wayness of the cipher and $\mathrm{Adv}_{\mathrm{QCCF}_{r,w}}^{\mathrm{IND}}(\mathcal{D})$ the advantage of adversary $\mathcal{D}$ challenging the indistinguishability of the key.

Best known attacks on underlying hard problems. Combining the results of [CS15] and [Sen11], the best known attacks on those problems have costs

$$\frac{\mathcal{WF}_{\mathcal{A}}(2r,r,t)}{\sqrt{n-k}} = 2^{1/2+t(1+o(1))-\log_2(2r)/2}$$

and

$$\frac{\mathcal{WF}_{\mathcal{A}}(2r,r,w)}{n-k} = 2^{1+w(1+o(1))-\log_2(2r)}$$

for any algorithm $\mathcal{A}$ among the variants of [Pra62; Ste88; Dum91; BJMM12; MMT11; MO15].

Number of bits of security. The number of bits of security of a problem is usually defined as the smallest λ such that for any adversaries $\mathcal{A}$

$$\frac{\mathrm{Time}(\mathcal{A})}{\mathrm{Adv}(\mathcal{A})} \geq 2^{\lambda}$$

where $\mathrm{Time}(\mathcal{A})$ is the running time of $\mathcal{A}$. Note that if $\mathcal{A}$ makes q queries to any oracle then $\mathrm{Time}(\mathcal{A}) > q$.

To summarize Theorem 2, in order to have λ bits of security against an IND-CCA adversary we want:

- $\mathcal{WF}_{\mathrm{QCSD}}(2r,r,t) > 2^{\lambda}$,
- $\mathcal{WF}_{\mathrm{QCCF}}(2r,r,w) > 2^{\lambda}$,
- $|\mathcal{M}| > 2^{\lambda}$,
- $\delta < 2^{-\lambda}$ *i.e.* $\mathrm{DFR}(\texttt{decoder}) < 2^{-\lambda}$.

Note that if all conditions are met except the last one on the DFR, the scheme is still IND-CPA secure.

Weak keys. Now, let us rewrite the definition of the decoding failure rate in the context of a Niederreiter cryptosystem with a sparse parity check matrix set $\mathcal{H}$ and the set of admissible error patterns $\mathcal{E}$. Let $\mathcal{W}$ be any subset (weak keys) of $\mathcal{H}$ and $\mathcal{D}$ be a decoder.

$$\begin{aligned}\mathrm{DFR}_{\mathcal{D}} &= \Pr\left[\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^{\top}) \neq \mathrm{e} \,\middle|\, \mathbf{H} \in \mathcal{H}, \mathrm{e} \in \mathcal{E}\right] \\ &= \frac{|\mathcal{W}|}{|\mathcal{H}|}\,\mathrm{DFR}_{\mathcal{D},\mathcal{W}} + \left(1 - \frac{|\mathcal{W}|}{|\mathcal{H}|}\right)\mathrm{DFR}_{\mathcal{D},\mathcal{H}\setminus\mathcal{W}}\end{aligned}$$

with

$$\mathrm{DFR}_{\mathcal{D},\mathcal{W}} = \Pr\left[\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^{\top}) \neq \mathrm{e} \,\middle|\, \mathbf{H} \in \mathcal{W}, \mathrm{e} \in \mathcal{E}\right]$$

and

$$\mathrm{DFR}_{\mathcal{D},\mathcal{H}\setminus\mathcal{W}} = \Pr\left[\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^{\top}) \neq \mathrm{e} \,\middle|\, \mathbf{H} \in \mathcal{H}\setminus\mathcal{W}, \mathrm{e} \in \mathcal{E}\right] .$$

In the context of decoding failures, a set of weak keys is a set $\mathcal{W}$ such that

$$\frac{|\mathcal{W}|}{|\mathcal{H}|}\,\mathrm{DFR}_{\mathcal{D},\mathcal{W}} > 2^{-\lambda} .$$

If this inequality is satisfied, but there is an imbalance between the factors so that $\frac{|\mathcal{W}|}{|\mathcal{H}|}$ is rather small but $\mathrm{DFR}_{\mathcal{D},\mathcal{W}}$ is rather large, then the estimates of the average DFR based on simulations could be skewed.

Error floor. Similarly, we can focus on the other side of the product $\mathbf{H}\mathrm{e}^\top$ and look at specific error patterns e. In this case, the phenomenon that we want to avoid is known as the error floor in coding theory. For all sparse parity check matrix $\mathbf{H}$, we associate $\mathcal{E}_\mathbf{H}$ a subset of $\mathcal{E}$. Let $\mathcal{D}$ be a decoder.

$$\mathrm{DFR}_\mathcal{D} = \frac{1}{|\mathcal{H}|}\sum_{\mathbf{H}\in\mathcal{H}}\left(\frac{|\mathcal{E}_\mathbf{H}|}{|\mathcal{E}|}\,\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}_\mathbf{H}} + \left(1-\frac{|\mathcal{E}_\mathbf{H}|}{|\mathcal{E}|}\right)\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}\setminus\mathcal{E}_\mathbf{H}}\right)$$

with

$$\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}_\mathbf{H}} = \Pr\left[\mathcal{D}(\mathbf{H},\mathbf{H}\mathrm{e}^\top)\neq \mathrm{e}\,\middle|\,\mathrm{e}\in\mathcal{E}_\mathbf{H}\right]$$

and

$$\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}\setminus\mathcal{E}_\mathbf{H}} = \Pr\left[\mathcal{D}(\mathbf{H},\mathbf{H}\mathrm{e}^\top)\neq \mathrm{e}\,\middle|\,\mathrm{e}\in\mathcal{E}\setminus\mathcal{E}_\mathbf{H}\right]\ .$$

If, for any sparse parity check matrix $\mathbf{H}$, we can find a set of error patterns $\mathcal{E}_\mathbf{H}$ that hinder the decoding performance, then this is problematic if

$$\frac{1}{|\mathcal{H}|}\sum_{\mathbf{H}\in\mathcal{H}}\frac{|\mathcal{E}_\mathbf{H}|}{|\mathcal{E}|}\,\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}_\mathbf{H}} = \mathrm{E}_{\mathbf{H}\in\mathcal{H}}\left[\frac{|\mathcal{E}_\mathbf{H}|}{|\mathcal{E}|}\,\mathrm{DFR}_{\mathcal{D},\mathbf{H}}^{\mathcal{E}_\mathbf{H}}\right] > 2^{-\lambda}\ .$$

Whether for weak keys or error floors, it is a matter of counting the problematic patterns and evaluating their influence on decoding performance.

3 A DFR extrapolation framework

Several works and simulation show that, excluding a phenomenon such as the error floor discussed in §5, the function $r\mapsto\log\mathrm{DFR}_\mathcal{D}(r)$ is always decreasing and, at worst, it is an affine function. Similar to how security reduction often relies on the complexity of the best known algorithm to solve a particular problem (*e.g.* information set decoding for most code-based cryptographic schemes), we make a decoding assumption that we can use to estimate the DFR of a particular decoding algorithm.

3.1 The decoder security assumption

In [SV19], a Markovian model was defined for a simple variant of bit-flipping: the step-by-step decoder. This decoder corrects fewer errors than other bit-flipping variants, however it uses the same ingredients: computing counters and flipping the corresponding positions if they are above some threshold. We fix the values of the parameters d and t, then a model can be generated and a DFR computed for arbitrary large values of the block size r. We observe that in the range of interest for r, the function $r\mapsto\log\mathrm{DFR}(r)$ is first strictly concave and eventually decreases linearly with r.

[Til18] also explores the asymptotic behaviour of QC-MDPC decoding. The asymptotic formula it provides for the DFR cannot be used directly because the setting is different (d and t vary with r), and also the conditions under which it can be proven are not relevant for decoders and parameters of practical interest. However, the indication provided by the formula is consistent with the previous remark, the dominant term in the expression decreases linearly with r.

Experimentally, when plotting the DFR (obtained by simulation) *vs.* the block size r in a logarithmic scale, one observes a concave curve. This was observed for all the variants of bit-flipping decoding.

As there is enough evidence from the state of the art to back this claim, we suggest formulating a decoding assumption on which we will base our estimation of the DFR. We denote $\mathcal{D}$ a decoder: it is a family of decoding algorithms that can be applied to any QC-MDPC codes with fixed (d, t) and variable r.

Assumption 1. For a given decoder $\mathcal{D}$, and a given security level λ, the function $r \mapsto \log \mathrm{DFR}_{\mathcal{D}}(r)$ is concave if $\mathrm{DFR}_{\mathcal{D}}(r) \geq 2^{-\lambda}$.

Under this assumption, a conservative estimation on the DFR can be made from the measure of the DFR for two different block sizes r_1 and r_2. Indeed, let $r_1 < r_2 < r_3$, then by definition of the concavity of a function we have

$$\log \mathrm{DFR}_{\mathcal{D}}(r_2) \geq \frac{r_3 - r_2}{r_3 - r_1} \log \mathrm{DFR}_{\mathcal{D}}(r_1) + \frac{r_2 - r_1}{r_3 - r_1} \log \mathrm{DFR}_{\mathcal{D}}(r_3)$$

as $\frac{r_3-r_2}{r_3-r_1}, \frac{r_2-r_1}{r_3-r_1} \in [0, 1]$ and $\frac{r_3-r_2}{r_3-r_1} r_1 + \frac{r_2-r_1}{r_3-r_1} r_3 = r_2$.

This inequality can be rewritten as in the following proposition.

Proposition 3. *Under Assumption 1, an upper bound of the DFR for a block size r_3 can be extrapolated from the DFR for block sizes r_1 and r_2, with $r_1 < r_2 < r_3$ using the following inequality:*

$$\log \mathrm{DFR}_{\mathcal{D}}(r_3) \leq \log \mathrm{DFR}_{\mathcal{D}}(r_1) + \frac{\log \mathrm{DFR}_{\mathcal{D}}(r_2) - \log \mathrm{DFR}_{\mathcal{D}}(r_1)}{r_2 - r_1} (r_3 - r_1) \, .$$

Another way to say it is that the secant line crossing the graph of $r \mapsto \log \mathrm{DFR}_{\mathcal{D}}(r)$ at r_1 and r_2 upper bounds $\log \mathrm{DFR}_{\mathcal{D}}(r_3)$ for any $r_3 > r_2$. It is an equality when $r \mapsto \log \mathrm{DFR}_{\mathcal{D}}(r)$ is affine, and this was the observed asymptotic behaviour in the Markovian model. However, in the range of values for r usually used in simulation, the function has a faster decrease and r_1 and r_2 should be as high as the simulation and the desired precision allow.

For instance in Figure 1, suppose the low curve (blue) is giving the $\log_2(\mathrm{DFR})$ and we are able to make accurate simulation as long as the DFR is above 2^{-25} (black dots). Taking the tangent at the last point gives us the red line from which we derive an upper bound r_{ext} for a block size with a DFR below 2^{-128}.

3.2 Confidence interval

The failure rate $\mathrm{DFR}_{\mathcal{D},\mathcal{H}}^{\mathcal{E}}(r)$ is the probability of the Bernoulli trial written in Algorithm 2 returning False. In this section we will denote this DFR as p for conciseness.

Algorithm 2: Bernoulli trial for decoding failure.

```
input : Block size $r$, message space $\mathcal{E}$, key space $\mathcal{H}$, decoder $\mathcal{D}$.
$\mathrm{e} \overset{\$}{\leftarrow} \mathcal{E}$;
$\mathbf{H} \overset{\$}{\leftarrow} \mathcal{H}$;
$\mathrm{s} \leftarrow \mathbf{H}\mathrm{e}^{\top}$;
$\mathrm{e}' \leftarrow \mathcal{D}(\mathbf{H}, \mathrm{s})$;
return $[\mathrm{e}' = \mathrm{e}]$;
```

Repeating this procedure N times and counting the number of failures F consitutes a random variable following a binomial distribution $\mathrm{Bin}(N, p)$. The ratio F/N gives an estimate of the actual probability p and, using a normal approximation, the standard deviation of this estimate is proportional to

$$\frac{\sqrt{F(N-F)}}{N\sqrt{N}}\,.$$

To be more precise, let us recall the definition of a confidence interval.

Definition 4. Let $\alpha \in (0,1)$ and suppose we have an observable vector $\mathrm{x} = (x_1, \dots, x_n)$ from a distribution that depends on an unknown parameter $\theta \in \Theta$. A *confidence interval* with a *confidence level* $(1-\alpha)$ for θ is obtained from a function $\mathcal{I}$ that satisfies:

$$\forall \theta \in \Theta, \quad \Pr\left[\theta \in \mathcal{I}(\mathrm{x}, \alpha) \,|\, \theta\right] \geq 1 - \alpha\,.$$

The actual value of the left-hand side of this inequality is called the *coverage probability.* So the confidence level is the infimum of the coverage probabilities for all $\theta \in \Theta$.

The aforementioned normal approximation is a common way to roughly estimate a confidence interval. Some more advanced techniques were designed so that the coverage probability is closer to the desired $(1-\alpha)$.

3.3 A first estimation

Here we will recall the Clopper-Pearson interval also known as the *exact* method as it gives a coverage probability that is always above $(1-\alpha)$. It is a conservative method for computing a confidence interval.

3.3.1 Clopper-Pearson interval

First, we need to set the framework for our problem. We repeat N times a Bernoulli process with probability of failure θ. The observed number of failures is a random variable that we denote $X \in \{0, \dots, N\}$. X follows a binomial distribution $\mathrm{Bin}(N, \theta)$:

$$\Pr[X = x] = \begin{cases} \binom{N}{x}\theta^x(1-\theta)^{N-x} & \text{if } x \in \{0, \dots, N\}; \\ 0 & \text{otherwise.} \end{cases}$$

The probability θ is the unknown parameter that we want to estimate.

This method uses the cumulative probabilities of the binomial distribution $\Pr[X \geq x]$ and $\Pr[X \leq x]$ for some number of failures x and some undeternmined probability θ:

$$\Pr[X \geq x] = \sum_{k=x}^{N} \binom{N}{k} \theta^k (1-\theta)^{N-k} \qquad \Pr[X \leq x] = \sum_{k=0}^{x} \binom{N}{k} \theta^k (1-\theta)^{N-k}\,.$$

The confidence interval is a range of values for θ such that for a number of failures F observed among N samples is not abnormal *i.e.* for any probability θ in the interval, the observation happens with probability at least $(1-\alpha)$.

Given the monotonicity of the cumulative probabilities with respect to θ, the confidence interval can be written as $(\theta_-;\theta_+)$ with

$$\theta_- = \inf_\theta \left\{ \theta \,\middle|\, \Pr[X \geq F] > \frac{\alpha}{2} \right\}, \qquad \theta_+ = \sup_\theta \left\{ \theta \,\middle|\, \Pr[X \leq F] > \frac{\alpha}{2} \right\}.$$

Another common and equivalent way to write it is as in the following proposition.

Proposition 5 (Clopper-Pearson [CP34]). *A $(1-\alpha)$-confidence interval for a failure probability p when F failures have been observed out of N experiments is (θ_-, θ_+) where*

$$\theta_- = B^{-1}(\alpha/2, F, N-F+1) \qquad \theta_+ = B^{-1}(1-\alpha/2, F+1, N-F)\,.$$

With B^{-1} the inverse of $x \mapsto B(x,a,b)$, the incomplete beta function:

$$B(x,a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1}\,dt\,.$$

3.3.2 A first estimation of confidence intervals for extrapolations

Now let us come back to our extrapolation problem. First, let us pretend that we know the true probabilities $p_1 = \mathrm{DFR}(r_1)$ and $p_2 = \mathrm{DFR}(r_2)$. If we suppose that $r \mapsto \log \mathrm{DFR}_{\mathcal{D}}(r)$ is an affine function (the extreme case of Assumption 1) then the failure probability $p_3 = \mathrm{DFR}(r_3)$ for a block size $r_3 > r_2 > r_1$ is

$$p_3 = p_1^{-A} p_2^{1+A} \tag{1}$$

with $A = \frac{r_3 - r_2}{r_2 - r_1} > 0$.

In reality, for a block size r_i, we only know that F_i failures were observed out of N_i samples with $i = 1, 2$. If p_1 has a $(1-\alpha)$-confidence interval $(p_1^-; p_1^+)$ and p_2 has a $(1-\beta)$-confidence interval $(p_2^-; p_2^+)$, then p_3 has a confidence interval $(p_3^-; p_3^+)$

$$p_3^- = (p_1^+)^{-A}(p_2^-)^{1+A}; \qquad p_3^+ = (p_1^-)^{-A}(p_2^+)^{1+A}$$

with a coverage probability at least $(1-\alpha)(1-\beta)$. This is illustrated in Figure 4.

However, for a given confidence level, a narrower confidence interval can be derived.

3.4 Using posterior probabilities

In this subsection we detail how to compute narrower confidence interval using posterior probabilities.

We see the failure probability of two Bernoulli trials as random variables $\Theta_i \in [0,1]$ for i$= 1, 2$. Under the condition that $\Theta_i = \theta_i$, we define X_i that follows a binomial distribution for $i = 1, 2$:

$$X_i \mid \Theta_i = \theta_i \sim \mathrm{Bin}(N_i, \theta_i)\,.$$

We define the random variable $\Theta_3 \in [0,1]$ as

$$\Theta_3 = \Theta_1^{-A} \Theta_2^{1+A}$$

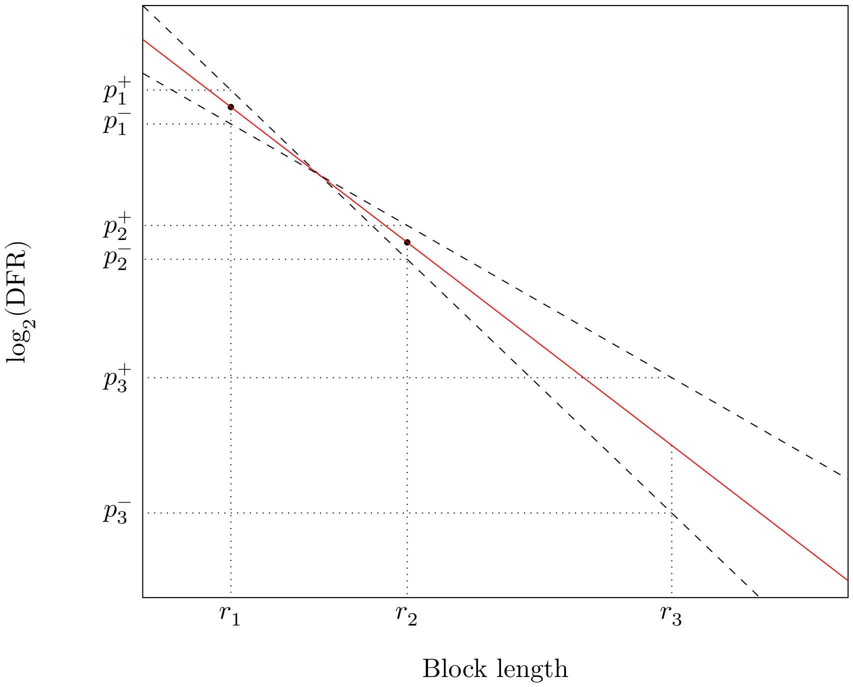

Figure 4: Rough estimate of the confidence interval.

for some constant $A > 0$.

Our goal is, with no prior knowledge on Θ_3, to find its distribution assuming that we observe $X_i = F_i$ for $i = 1, 2$.

Let us first recall two properties about random variables (see [Spr79; BT02] for proofs). In this subsection, we denote by f_X the probability density functions of any random variable X.

Proposition 6. *Let X and Y be two independent positive random variables whose probability density functions are respectively f_X and f_Y then the ratio $Z = X/Y$ is a random variable with the following probability density function:*

$$f_{X/Y}(z) = \int_0^\infty f_Y(y) f_X(zy) y \, dy \, .$$

Proposition 7. *Let X be a real valued random variable and let $g : \mathbb{R} \to \mathbb{R}$ be an invertible monotonic differentiable measurable function whose inverse is h. Let Y be the random variable defined by $Y = g(X)$. We write f_X and f_Y the probability density functions of respectively X and Y. Then*

$$f_Y(y) = |h'(y)| f_X(h(y)) \, .$$

Corollary 8. *For any random variable X and any real number $A \neq 0$*

$$f_{X^A}(x) = \left|\frac{1}{A}\right| x^{1/A-1} f_X(x^{1/A}) \, .$$

With no prior knowledge on the failure probability distributions we have the following proposition.

Proposition 9. *For $i = 1, 2$, taking the uniform prior $f_{\Theta_i}(\theta_i) = 1$, under the observation $X_i = x_i$, Θ_i follows a beta distribution $\mathrm{B}(x_i + 1, N_i - x_i + 1)$:*

$$f_{\Theta_i \,|\, X_i = x_i}(\theta) = \frac{1}{B(x_i + 1, N_i - x_i + 1)} \theta^{x_i} (1 - \theta)^{N_i - x_i}$$

where $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

Proof. Bayes' theorem states that:

$$f_{\Theta_i \,|\, X_i = x_i}(\theta) = \frac{\Pr(X_i = x_i \,|\, \Theta_i) f_{\Theta_i}(\theta)}{\int_0^1 \Pr(X_i = x_i \,|\, \Theta_i) f_{\Theta_i}(\theta) d\theta} . \qquad \square$$

Applying Corollary 8, we can determine the distributions of Θ_1^A and Θ_2^{1+A}. Then, applying Proposition 6 we obtain

$$\begin{aligned} & f_{\Theta_3 \,|\, X_1 = x_1, X_2 = x_2}(z) \\ & = \frac{z^{1/(1+A) - 1}}{A(1 + A)} \int_0^1 f_{\Theta_1 \,|\, X_1 = x_1}(x^{1/A}) f_{\Theta_2 \,|\, X_2 = x_2}((zx)^{1/(1+A)}) x^{1/A + 1/(1+A) - 1} \, dx \,. \end{aligned}$$

Finally, since the integrand is non-negligible only in a narrow range of values and since we are interested in the logarithm of the DFR, we can apply a change of variable and Proposition 7 to obtain the following proposition.

Proposition 10. *Let $\Theta_i \in [0, 1]$ be a real-valued random variable for $i= 1, 2$. Let X_i be random variables such that under the condition that $\Theta_i = \theta_i$, X_i follows a binomial distribution for $i = 1, 2$:*

$$X_i \,|\, \Theta_i = \theta_i \sim \mathrm{Bin}(N_i, \theta_i) \,.$$

Let $\Theta_3 \in [0, 1]$ be the random variable defined as

$$\Theta_3 = \Theta_1^{-A} \Theta_2^{1+A}$$

for some constant $A > 0$.

We have

$$\begin{aligned} & f_{\log(\Theta_3) \,|\, X_1 = x_1, X_2 = x_2}(s) \\ & \quad = \frac{e^{s(x_2+1)/(1+A)}}{K} \int_{-\infty}^0 e^{t(\frac{x_1+1}{A} + \frac{x_2+1}{1+A})} (1 - e^{\frac{t}{A}})^{N_1 - x_1} (1 - e^{\frac{s+t}{1+A}})^{N_2 - x_2} \, dt \end{aligned}$$

with

$$K = A(1 + A) B(x_1 + 1, N_1 - x_1 + 1) B(x_2 + 1, N_2 - x_2 + 1) \,.$$

We can now define the $(1 - \alpha)$-confidence interval $(\log \theta_3^-; \log \theta_3^+)$ on the extrapolated value $\log \Theta_3$ where $\log \theta_3^-$ is such that

$$\int_{-\infty}^{\log \theta_3^-} f_{\log(\theta_3)}(s) \, ds = \alpha/2$$

and $\log \theta_3^+$ is such that

$$\int_{\log \theta_3^+}^0 f_{\log(\theta_3)}(s) \, ds = \alpha/2 \,.$$

Proposition 11. *Let p_1 and p_2 be the failure probabilities of two independent Bernoulli trials. Suppose, for $i = 1, 2$, that we observe F_i failures out of N_i samples of each Bernoulli trial, and suppose that*

$$p_3 = p_1^{-A} p_2^{1+A} .$$

Then a $(1-\alpha)$-confidence interval for p_3 is $(\exp(\ell p_3^-), \exp(\ell p_3^+))$ where ℓp_3^- is the greatest value such that

$$\frac{1}{K} \int_{-\infty}^{\ell p_3^-} \int_{-\infty}^{0} e^{s\left(\frac{F_2+1}{1+A}\right)} e^{t\left(\frac{F_1+1}{A}+\frac{F_2+1}{1+A}\right)} (1-e^{\frac{t}{A}})^{N_1-F_1} (1-e^{\frac{s+t}{1+A}})^{N_2-F_2} \, dt \, ds < \frac{\alpha}{2}$$

and ℓp_3^+ is the smallest value such that

$$\frac{1}{K} \int_{\ell p_3^+}^{\infty} \int_{-\infty}^{0} e^{s\left(\frac{F_2+1}{1+A}\right)} e^{t\left(\frac{F_1+1}{A}+\frac{F_2+1}{1+A}\right)} (1-e^{\frac{t}{A}})^{N_1-F_1} (1-e^{\frac{s+t}{1+A}})^{N_2-F_2} \, dt \, ds < \frac{\alpha}{2}$$

where

$$K = A(1+A)B(F_1+1, N_1-F_1+1)B(F_2+1, N_2-F_2+1) .$$

Example 12. In Table 1 we compare the confidence intervals $(\mathrm{DFR}^-, \mathrm{DFR}^+)$ for extrapolated DFR at $r_3 = 12\,323$ for two simulations data:

- BGF (7 iterations): for $r_1 = 10\,037$, $F_1 = 66\,391$, $N_1 = 3\,747\,161\,784$, $r_2 = 10\,253$, $F_2 = 5$, $N_2 = 1\,445\,221\,866$,
- Backflip 7 iterations : $r_1 = 10\,181$, $F_1 = 394$, $N_1 = 14\,576\,092\,619$, $r_2 = 10\,253$, $F_1 = 111$, $N_1 = 34\,283\,154\,045$.

We can see a difference of a few bits between confidence intervals from the simple method and Proposition 11.

		Simple method		Proposition 11	
Algorithm	$\log_2(\mathrm{DFR})$	$\log_2(\mathrm{DFR}^-)$	$\log_2(\mathrm{DFR}^+)$	$\log_2(\mathrm{DFR}^-)$	$\log_2(\mathrm{DFR}^+)$
BGF 7 it.	-146.20	-172.30	-129.11	-164.21	-130.31
Backflip 7 it.	-116.22	-134.13	-99.00	-128.13	-104.57

Table 1: Confidence intervals for some decoding simulation data.

4 Weak keys: Subsets of parity check matrices

Short cycles in the Tanner graph and high number of intersections between columns in the parity check matrix are detrimental for the decoding capabilities of the decoder.

In this section, we start from the definition of the distance spectrum due to [GJS16]. We use properties of this spectrum from §A that can be used to enumerate them. We make an observation on the counters distribution for parity check matrices with unusual spectrum and see how that can be detrimental to the decoding performance. Then, we construct three sets of weak keys for a QC-MDPC system, these keys have an unusual spectrum and thus a DFR higher than average. We eventually assess their impact on the security of the system by estimating their DFR with simulation and using upper bounds on their cardinality.

4.1 QC-MDPC Codes

Definition and polynomial representation. Let $\mathbf{H} = (\mathbf{H}_0, \mathbf{H}_1)$ be the parity check matrix of a QC-MDPC code. Equivalently, the parity check matrix can be written as a tuple of polynomials $(\mathrm{h}_0, \mathrm{h}_1) \in (\mathbb{F}_2[x]/(x^r - 1))^2$, using the following isomorphism.

Proposition 13 (Recall)**.** *The application*

$$\mathbf{H} \mapsto h_{0,0} + h_{1,0}x + \cdots + h_{r-2,0}x^{r-2} + h_{r-1,0}x^{r-1}$$

is an isomorphism between the ring of circulant $r \times r$ matrices with coefficients in $\mathbb{F}_2$ and the quotient ring $\mathbb{F}_2[x]/(x^r - 1)$.

If we denote h_i the polynomial corresponding to first column of the block $\mathbf{H}_i$ for $i = 0, 1$, and if we write the row vector $(\mathrm{e}_0, \mathrm{e}_1)$ as polynomials then the product

$$(\mathbf{H}_0, \mathbf{H}_1)(\mathrm{e}_0, \mathrm{e}_1)^\top$$

is the column vector represented by the polynomial $\mathrm{e}_0\mathrm{h}_0 + \mathrm{e}_1\mathrm{h}_1$.

The following isomorphism is important in our analysis as it allows us to reduce properties on any distance δ to the case $\delta = 1$.

Proposition 14. *For all $\delta \in \mathbb{Z}_r^\times$, the endomorphism ϕ_δ of $(\mathbb{F}_2[x]/(x^r - 1), +, \times)$ induced by*

$$x \mapsto x^\delta$$

is an isomorphism and an isometry for the Hamming distance.

Remark 15. The inverse of ϕ_δ is $\phi_{\delta^{-1}}$.

Remark 16. Previous works on weak keys for QC-LDPC or QC-MDPC codes also mention this isometry: [APRS20, §4.3] & [BDLO16, §6]. Together with the circular shifts (multiplications by x^i for all $i \in \{0, \dots, r-1\}$), it allows a quadratic gain on the size of a set of keys while holding similar characteristics.

Decoding. To understand the rationale behind our constructions of weak keys, let us first adopt an intuitive point of view of the bit-flipping decoder. Remember Algorithm 1, the bit-flipping algorithm. The counter $|\mathbf{h}_j \star \mathrm{s}|$ of a position j is the number of parity check equations (rows of $\mathbf{H}$) involving that position and which are unsatisfied. If the number of unsatisfied parity check equations is high, the coordinate on that position is likely to be erroneous.

Finding e from the syndrome $\mathrm{s} = \mathbf{H}\mathrm{e}^\top$ and the sparse parity check matrix $\mathbf{H}$ such that $|\mathrm{e}| = t$ is possible by exploiting the bias in the counters. Figure 5 gives the number of positions with given counter values, the smaller Gaussian-shaped curve on the right represents the erroneous positions. The decoder knows the counters but not the errors' location (*i.e.* it knows only the sum of the two

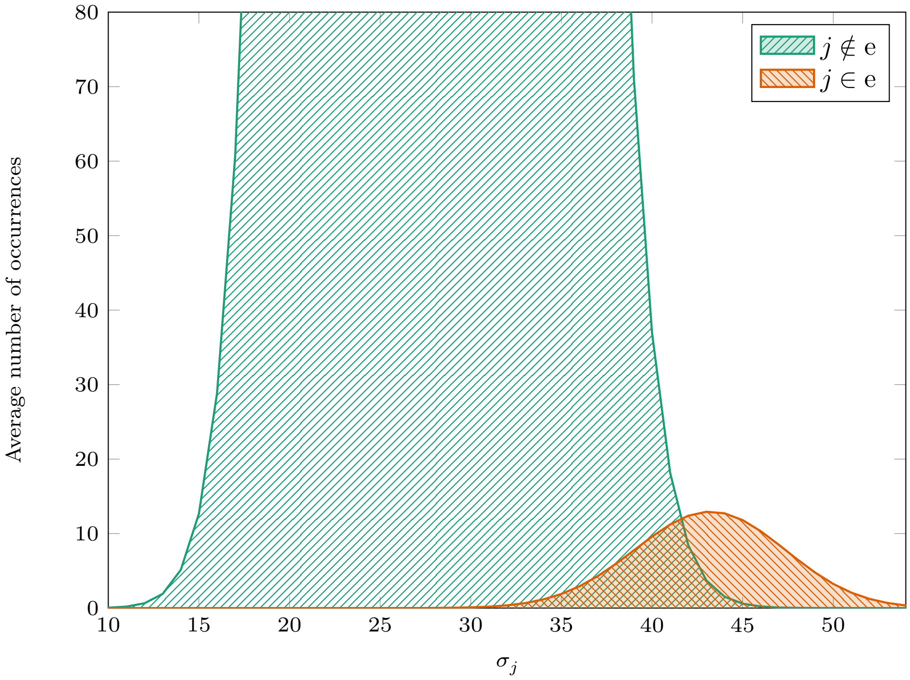

Figure 5: Counters distributions for $(r, d, t) = (12\,323, 71, 137)$.

curves). So Algorithm 1 chooses a sensible threshold T and flips all positions with a counter above it, the syndrome is recomputed, then the counters again. This process is repeated and the error weight usually decreases after each iteration until all errors have been removed and the syndrome is zero.

Remark 17. When the block size decreases or the error weight increases, this usually results in a decrease of the error counters. When this happens, the small Gaussian curve in Figure 5 will move to the left and error detection will become more difficult—even if the threshold is adjusted—because they will be overwhelmed by the sheer mass of correct positions. Any effect that moves the small curve to the left or the large curve to the right therefore has a negative effect on the decoder behaviour. As we will see later, this is precisely what happens with weak keys.

4.2 Weak keys: Constructions and properties

IND-CCA security and weak keys for KEMs. Remember that a KEM (KeyGen, Encaps, Decaps) is said to be δ-correct if

$$\Pr\left[\mathrm{Decaps}(\mathtt{sk}, c) \neq K \,\middle|\, (\mathtt{pk}, \mathtt{sk}) \leftarrow \mathrm{KeyGen}(\{0,1\}^\lambda); (K, c) \leftarrow \mathrm{Encaps}(\mathtt{pk})\right] \leq \delta\,.$$

And remember that for [BIKE], since the encapsulation includes a randomization, δ is the average DFR for a specific decoder, for any key and any error vector.

We can summarize Theorem 2 in the case of BIKE, for any IND-CCA adversary $\mathcal{B}$ against BIKE issuing at most q queries to any random oracle with a decoder $\mathcal{D}$ for a set of keys $\mathcal{H}$, with

$$\mathrm{Adv}_{\mathrm{BIKE}}^{\text{IND-CCA}}(\mathcal{B}) \leq q \cdot \mathrm{DFR}_{\mathcal{D},\mathcal{H}} + \epsilon$$

where ϵ encompasses the advantage related to the underlying difficult problems and therefore does not depend at all on the decoder used for the system. This ϵ is dealt with in the usual manner: proper semantically secure transform and parameters selection according to the computational assumptions.

If one wants to challenge the fact that the decoder actually offers the required security, *i.e.* its failure rate is less than $2^{-\lambda}$, one could present a set of weak keys $\mathcal{W} \subset \mathcal{H}$. The average DFR for these keys $\mathrm{DFR}_{\mathcal{D},\mathcal{W}}$ would have to be higher than $\mathrm{DFR}_{\mathcal{D},\mathcal{H}}$ but its density would also have to be high enough to contribute significantly to the average DFR, *i.e.* the set $\mathcal{W}$ has to be such that

$$\frac{|\mathcal{W}|}{|\mathcal{H}|} \mathrm{DFR}_{\mathcal{D},\mathcal{W}} > 2^{-\lambda} .$$

In the following paragraphs we will construct three categories of weak keys and evaluate the left-hand side of the above inequality.

Type I. In [DGK19], weak keys are specified as $(\mathrm{h}_0, \mathrm{h}_1)$ with

$$\mathrm{h}_0 = (1 + x + ... + x^{f-1}) + \mathrm{h}_0'$$

such that $|\mathrm{h}_0'| = d - f$ and $\left|(1 + x + ... + x^{f-1}) \star \mathrm{h}_0'\right| = 0$ for f in a range from 0 to 40. Authors observe that the correcting capability deteriorates as f grows. Values as high as 40 always lead to a decoding failure in simulation.

The reason for this degradation comes from the fact that, compared to a typical key, a weak key admits column pairs with a larger intersection in its private parity check matrix. This can be seen by computing the spectral polynomial[3] of h_0:

$$\begin{aligned} \mathrm{s}(\mathrm{h}_0) = \mathrm{h}_0 {\mathrm{h}_0}^{\top} &= (1 + x + ... + x^{f-1})(1 + x^{-1} + ... + x^{-(f-1)}) + \mathrm{s}' \\ &= f + (f-1)x + ... + x^{f-1} + \mathrm{s}' \end{aligned}$$

where s' has nonnegative coefficients. So any column $x^j \mathrm{h}_0$ has at least $(f-1)$ intersections with its two neighbours $x^{j \pm 1}\mathrm{h}_0$, at least $(f-2)$ intersections with $x^{j \pm 2}\mathrm{h}_0$, *etc.*

The typical maximum column intersection of BIKE keys is small: for $(r, d) = (12\,323, 71)$ about a quarter of the keys have a maximal column intersection greater than 5 (see Table 5). More intersections between columns mean higher correlations between their counters. In Figure 6 we measure the difference between the weak keys defined above with parameter $f = 20$ and random keys. With a weak key, an erroneous position tends to have lower counter when its immediate neighbour is erroneous. Conversely, still with a weak key, a non-erroneous position tends to have higher counter when its neighbour is erroneous.

Intuitively, this means that (i) neighbours that are both erroneous tend to hide each other and (ii) an erroneous position will contaminate its correct

[3] See Definition 30 in §A

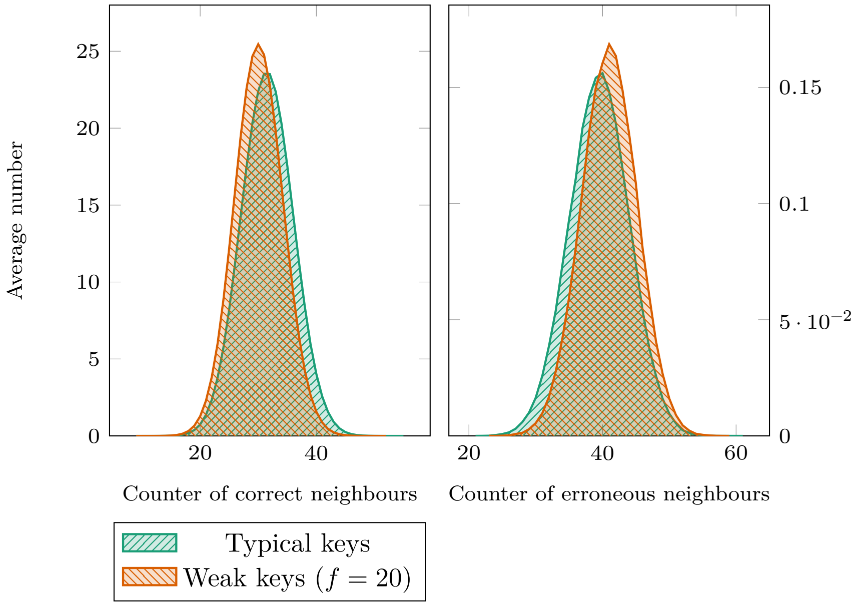

Figure 6: Counter values of the neighbours of an error for typical keys and weak keys with $f = 20$.
$(r, d, t) = (12\,323, 71, 134)$.

neighbours. Both effects negatively impact the (threshold-based) decoder (see Remark 17). Indeed, a higher counter on average for correct positions implies that more of them are above the threshold and are thus being flipped, adding errors. A lower counter for errors implies that more of them are below the threshold and are thus left unchanged, not decreasing the error weight.

As the decoding degradation is explained by the abnormal distribution of multiplicities in the spectrum of a block, we can generalize the construction of the weak keys. Using Remark 33 and Proposition 34, from one key defined as in [DGK19] we derive many more with the same multiplicity distribution (up to a permutation of the distance values in the spectrum).

Definition 18. We call weak key of Type I and parameter f, a key $\mathrm{h} = (\mathrm{h}_0, \mathrm{h}_1)$ such that

$$\mathrm{h}_i = \phi_\delta(x^\ell(1 + x + \ldots + x^{f-1}) + \mathrm{h}'_i)$$

for some $i \in \{0, 1\}$, $\ell \in \{0, \ldots, r-1\}$, with $|\mathrm{h}'_i| = d - f$ and $|\mathrm{h}_i| = d$.

Algorithm 3 gives a generation algorithm for Type I weak keys.

Proposition 19. *We denote $\mathcal{W}_I(f)$ the set of weak keys of Type I of parameter f with blocks of weight d and length r.*

$$|\mathcal{W}_I(f)| \le 2r \left\lfloor \frac{r}{2} \right\rfloor \binom{r-f}{d-f}.$$

The construction of weak keys of type I in Definition 18 affects the multiplicity of many distances: $\mu\left(\delta, \mathrm{h}_i\right) \ge f - 1$; $\mu\left(2\delta, \mathrm{h}_i\right) \ge f - 2$; $\ldots$; $\mu\left((f-1)\delta, \mathrm{h}_i\right) \ge 1$.

Algorithm 3: Type I weak keys generation.

```
input  : Block size $r$, column weight $d$, an integer $f$.
output : $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ with $|\mathrm{h}| = d$ and $f$ $\delta$-consecutive positions.
$(p_1, p_2, \dots, p_f) \leftarrow (0, 1, \dots, f-1)$;
Sample $(d-f)$ values $(p_{f+1}, \dots, p_d)$ from $\{f, \dots, r-1\}$;
$\delta \overset{\$}{\leftarrow} \{1, \dots, \lfloor r/2 \rfloor\}$;
$\ell \overset{\$}{\leftarrow} \{0, \dots, r-1\}$;
$\mathrm{h} \leftarrow 0$;
for $k \in \{1, \dots, d\}$ do
    /* Coordinate transformation to directly compute $\phi_\delta(x^\ell \mathrm{h})$.
       */
    $h_{\delta(\ell+p_k)} \leftarrow 1$;
return h;
```

Type II. Instead of having several high multiplicities at the same time, Type II weak keys only increase the multiplicity of a single distance. We will see that they have a lower impact on the DFR for a given multiplicity, but in return a higher density.

Definition 20. We call weak key of Type II and parameter m, a key $\mathrm{h} = (\mathrm{h}_0, \mathrm{h}_1)$ such that $\mu(\delta, \mathrm{h}_i) = m$ for some $i \in \{0, 1\}$ and some distance $\delta \in \{1, \dots, \lfloor r/2 \rfloor\}$.

Thanks to Corollary 41 of §A we may obtain an upper bound for the number of Type II weak keys.

Proposition 21. *Let m be an integer such that $0 \le m < d$. We denote $\mathcal{W}_{II}(m)$ the set of patterns* h *of weight d and length r for which one distance at least of its spectrum has multiplicity m. Then*

$$|\mathcal{W}_{II}(m)| \le 2 \left\lfloor \frac{r}{2} \right\rfloor \frac{r}{d-m} \binom{d-1}{d-m-1} \binom{r-d-1}{d-m-1} .$$

For a given m, the sets $\{\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1) \mid \mu(\delta, \mathrm{h}) = m\}$, for all possible values of δ, can intersect. Therefore we only give an upper bound. But in practice, when m is above the typical values (4 or 5), the intersections are very small and the bound is very tight. Algorithm 4 derives from the combinatorial analysis of §A and gives a generation algorithm for Type II weak keys. Its correctness is guaranteed by Proposition 40 and Corollary 32.

Remark 22. With BIKE, the suggested decoders are parallel: the syndrome is only computed once for each iteration and flips are chosen independently of the order in which positions are considered in an iteration. Therefore, the decoders are such that

$$\mathcal{D}((\phi_\delta(\mathrm{h}_0), \phi_\delta(\mathrm{h}_1)), \phi_\delta(\mathrm{s})) = \phi_\delta(\mathcal{D}((\mathrm{h}_0, \mathrm{h}_1), \mathrm{s})) .$$

This means that the set of patterns for which a distance δ has a multiplicity m has the same DFR as the set of patterns for which a distance $\delta' \neq \delta$ has a multiplicity m. Put another way, for a given multiplicity m, for all distances $\delta \in \{1, \dots, \lfloor r/2 \rfloor\}$ the constructed sets have exactly the same contribution to the average DFR.

Algorithm 4: Type II weak keys generation.

```
input  : Block size $r$, column weight $d$, an integer $m$.
output : $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ with $|\mathrm{h}| = d$ and $\exists \delta, \mu(\delta, \mathrm{h}) = m$.
$s \leftarrow d - m$;
$a_0 \leftarrow 0$; $a_s \leftarrow d$;
$b_0 \leftarrow 0$; $b_s \leftarrow r - d$;
Sample $(s-1)$ values $(a_1, \dots, a_{s-1})$ from $\{1, \dots, d-1\}$;
Sample $(s-1)$ values $(b_1, \dots, b_{s-1})$ from $\{1, \dots, r-d-1\}$;
/* Componentwise subtraction.                                        */
$(o_1, \dots, o_s) \leftarrow (a_1, \dots, a_s) - (a_0, \dots, a_{s-1})$;
$(z_1, \dots, z_s) \leftarrow (b_1, \dots, b_s) - (b_0, \dots, b_{s-1})$;
$\delta \overset{\$}{\leftarrow} \{1, \dots, \lfloor r/2 \rfloor\}$;
$\ell \overset{\$}{\leftarrow} \{0, \dots, z_1 + o_1 - 1\}$;
$\mathrm{h} \leftarrow 0$;
$i \leftarrow -\ell$;
for $j \in \{1, \dots, s\}$ do
    $i \leftarrow i + z_j$;
    for $k \in \{0, \dots, o_j - 1\}$ do
        /* Coordinate transformation to directly compute
            $\phi_\delta(x^\ell \mathrm{h})$.                                */
        $h_{\delta(i+k)} \leftarrow 1$;
    $i \leftarrow i + o_j$;
return h;
```

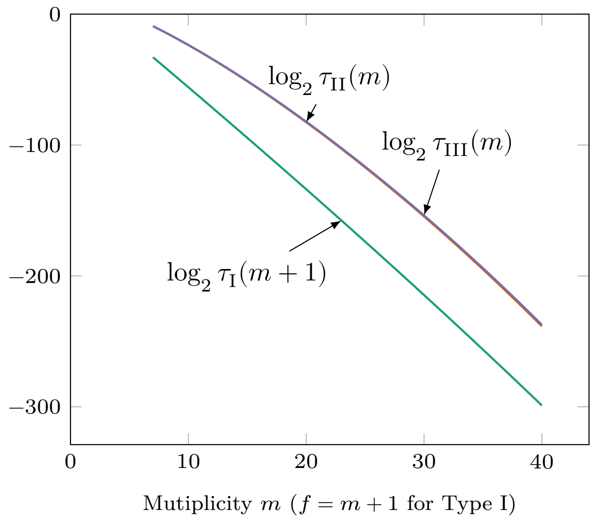

Figure 7: Density of weak keys versus multiplicity (log scale) for $(r, d) = (12\,323, 71)$. Type III keys are slightly denser, but their count is very close to the count for Type II.

Type III. Weak keys of Type I and II have properties that concern only one block of the parity check matrix. We can also define weak keys that have many intersections between two columns of two different blocks.

Definition 23. We call weak key of Type III and parameter m, a key $\mathrm{h} = (\mathrm{h}_0, \mathrm{h}_1)$ such that $\left|\mathrm{h}_0 \star x^\ell \mathrm{h}_1\right| = m$ for some $\ell \in \{0, \dots, r-1\}$.

Proposition 24. *We denote $\mathcal{W}_{III}(m)$ the set of weak keys of Type III of parameter m with blocks of weight d and length r.*

$$|\mathcal{W}_{III}(m)| \leq r \binom{d}{m} \binom{r-d}{d-m}$$

Statistics. In Figure 7 we give the density of all types of weak keys for $(r, d) = (12\,323, 71)$. We denote for all $m > 0$ and for type $\in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$

$$\tau_{\mathrm{type}}(m) = \frac{\left|\mathcal{W}_{\mathrm{type}}(m)\right|}{\binom{r}{d}}\,.$$

We shift the Type I curve ($m = f - 1$) to align the multiplicities. As we will observe in §4.3, the Type I keys have a worse effect on decoding for a given multiplicity. We observe also that for large multiplicity (roughly above $f = 21$ for Type I, and above $m = 27$ for Type II and III) the density is small enough to make those keys harmless (assuming a target of 128 bits of security), regardless of their impact on decoding.

4.3 DFR estimations

To estimate the DFR for weak keys, we will rely on the framework developed in §3 based on Assumption 1. To be more precise, we assume in this subsection that for any subset of keys $\mathcal{W} \subset \mathcal{H}$, the function $r \mapsto \mathrm{DFR}_{\mathcal{D},\mathcal{W}}(r)$ is concave. In other words, the high distance multiplicity will not have a stronger effect when the block size gets larger, but will affect the decoder similarly for any block size.

In §B, we give simulation results for all the types of weak keys previously defined with several values for f or m using BGF with 5 iterations (7 syndrome calculations). We can see that no set of weak keys seems to be any threat as the product of their density by their average DFR is always below $2^{-\lambda}$, where $\lambda \in \{128, 192, 256\}$ is the security parameter.

For Level 1 parameters, we can observe that Type I keys with $f \geq 10$ and Type II and III with $m \geq 14$ have a negligible influence on the average. For the other weak keys, with lower mutiplicity, the estimated DFR for weak keys is within the confidence interval of the average DFR obtained with random keys. This means that for $f < 10$ and $m < 14$ we did not observe in our experiment a measurable difference in the decoder's DFR between weak keys and random keys.

While all types of weak keys that we established have at least a pair of columns with $(f-1)$ (for Type I) or m (for Type II or III) intersections, some differences explain the different DFR. In a weak key of Type I and parameter f, a column $x^j \mathrm{h}_i$ has at least $(f-1)$ intersections with its two neighbours $x^{j\pm\delta}\mathrm{h}_i$, at least $(f-2)$ intersections with $x^{j\pm2\delta}\mathrm{h}_i$, etc. In a weak key of Type II and parameter m, a column $x^j \mathrm{h}_i$ has exactly m intersections with its two neighbours $x^{j\pm\delta}\mathrm{h}_i$. And in a weak key of Type III and parameter m, a column $x^j \mathrm{h}_0$ has exactly m intersections with a single column $x^{j'}\mathrm{h}_1$.

5 Error floors: Subsets of error patterns

LDPC codes and Turbo codes are known to suffer from a phenomenon called error floor (see [Gar+01; Ric03]). When the signal-to-noise ratio increases, the decoding performance of these codes first undergoes a sharp decrease in a region called the waterfall region, then there is a sudden change in slope and the performance flattens out (see Figure 1). This latter phenomenon is called the error floor.

Just like LDPC codes, QC-MDPC codes are not spared from error floor phenomena. However, the techniques used to estimate the error floor of LDPC codes usually rely on enumerating subgraphs of the Tanner graph of the code. The Tanner graph is denser for an MDPC code and these methods are usually not practical.

To initiate the discussion of the techniques used in this section, let us first consider the case of a minimum distance decoder.

Minimum distance decoding. We assume that we decode an error pattern of weight t in a QC-MDPC code with $\mathcal{D}$ a minimum distance decoder (*i.e.* it outputs the smallest possible error pattern e which, when added to the input, gives a codeword).

Recall that we have defined the DFR as the probability that the decoder outputs an error vector that is different from the one that was used to compute the syndrome:

$$\mathrm{DFR}_{\mathcal{D},\mathcal{H}}^{\mathcal{E}} = \Pr\left[\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^\top) \neq \mathrm{e} \,\middle|\, \mathbf{H} \in \mathcal{H}, \mathrm{e} \in \mathcal{E}\right] .$$

With a minimum distance decoder, we have a decoding failure $\mathcal{D}(\mathbf{H}, \mathbf{H}\mathrm{e}^\top) \neq \mathrm{e}$ when

$$\mathrm{e} = \mathrm{c} + \mathrm{e}' \qquad \text{or equivalently} \qquad \mathrm{e}' = \mathrm{c} + \mathrm{e} \tag{2}$$

with c a codeword and $|\mathrm{e}'| \leq t$.

We know that

$$|\mathrm{e}'| = |\mathrm{c}| + |\mathrm{e}| - 2\,|\mathrm{e} \star \mathrm{c}| \ ,$$

which means that

$$\frac{|\mathrm{c}|}{2} \leq |\mathrm{e} \star \mathrm{c}| \ .$$

If e, e′ and c are taken uniformly at random, the probability for this inequality to be satisfied is

$$\sum_{i=\lceil |\mathrm{c}|/2 \rceil}^{\min(|\mathrm{c}|,t)} \frac{\binom{|\mathrm{c}|}{i}\binom{n-|\mathrm{c}|}{t-i}}{\binom{n}{t}} \ . \tag{3}$$

If we write the parity check matrix $\mathbf{H} \in \mathcal{H}$ of a QC-MDPC as

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \end{pmatrix}$$

then we have a generator matrix $\mathbf{G} \in \mathcal{H}$

$$\mathbf{G} = \begin{pmatrix} \mathbf{H}_1^\top & \mathbf{H}_0^\top \end{pmatrix} \ .$$

So, taking each row of $\mathbf{G}$, we see that there are r codewords of weight w, they are all obtained by circularly shifting the first row. As w is well below the Gilbert-Varshamov bound, it is safe to assume that there are no smaller codewords in the general case.

Moreover, most of the parameters used to instantiate a QC-MDPC system will take into account the slight asymmetry in the cost of the key and message attacks respectively (because of [Sen11]) by having $w > t$. Therefore, in (3), since the sum is zero when $w > 2t$, the sum of two codewords with weight w will probably not trigger a decoding failure because they have a weight slightly less than $2w$. In the end we are only concerned by the r codewords of weight w.

Given the previous remark and the fast decay in the summand when i grows in (3), we can approximate the probability of having a bad error pattern with

$$r \frac{\binom{w}{\lceil w/2 \rceil}\binom{n-w}{t-\lceil w/2 \rceil}}{\binom{n}{t}} \ . \tag{4}$$

In order to put into perspective the rarity of such a configuration, let us give an asymptotic formula for (4) when $r \to \infty$:

$$2^{A - B \log_2(r)} \tag{5}$$

where

$$A = \log_2 \left(\binom{w}{\lceil w/2 \rceil} \frac{t!}{(t - \lceil w/2 \rceil)!} \right) - \left\lceil \frac{w}{2} \right\rceil \quad \text{and} \quad B = \left(\left\lceil \frac{w}{2} \right\rceil - 1 \right) \ .$$

Numerical application of this formula are given in Table 2. We can see that it is always negligible compared to $2^{-\lambda}$ where λ is the security parameter.

λ	w	t	A	B	r	$A - B\log_2(r)$
128	142	134	535.496	70	12 323	−415.738
192	206	199	838.289	102	24 659	−649.874
256	274	264	1 171.659	136	40 597	−910.376

Table 2: Numerical application for (5) with BIKE parmeters.

In this section we will see that in practice, as the decoder is not minimum distance and is typically a bit-flipping decoder, we can slightly relax the construction of e' in (2). Indeed, we will see that c need not necessarily be a codeword (it can be a near-codeword, see Definition 26) and that we can choose $|\mathrm{e}'| > t$. Although these new conditions do not necessarily trigger a decoding failure, they do have a negative impact on decoding performance to an extent that we will quantify using simulations.

In a QC-MDPC code, the quasi-cyclicity of the code endows it with a polynomial structure. Together with the fact that by definition it has a sparse parity check matrix, and it is defined in a field of characteristic 2, we shall see that many near-codewords can be found. We will build subsets of error patterns that are spheres around codewords or near-codewords for a certain radius. We will see that, using a bit-flipping decoder, they have a higher DFR. The density of these subsets multiplied by their DFR is a lower bound of the average DFR. Since the slope is rather flat when r increases, we refer to the curve of this lower bound as the error floor.

5.1 Structured patterns in QC-MDPC codes

Low weight codewords. There is an isometry between codewords of a QC-MDPC code of rate 1/2 and those of its dual. The following definition describes the set of codewords of weight w for a QC-MDPC.

Definition 25. Let $\mathrm{h} = (\mathrm{h}_0, \mathrm{h}_1) \in (\mathbb{F}_2[x]/(x^r - 1))^2$ be the parity check matrix of a QC-MDPC code of row weight w.

We write $\mathcal{C}$ the set of codewords of weight w:

$$\mathcal{C} = \{(x^s\mathrm{h}_1, x^s\mathrm{h}_0) \mid s \in \{0, \dots, r-1\})\} \ .$$

The sum of any two codewords of weight w gives the set of codewords $2\mathcal{C}$. And any $c \in 2\mathcal{C}$ has weight $2w - \epsilon_0 - \epsilon_1$ where the distribution of ϵ_i is given by π_m in Corollary 44 from §A for $m = \epsilon_i$ and $i = 0, 1$. These codewords have too great a weight to be relevant here. Indeed, we will construct error patterns of weight t that are close to codewords. With the usual BIKE parameters, we already have $w > t$, doubling the size of the codewords can only increase the distance to the error patterns of fixed weight t.

Near-codewords.

Definition 26. Let $\mathbf{H}$ be the parity check matrix of a linear code. A (u, v) near-codeword is an error pattern e of weight u such that $\left|\mathbf{H}\mathrm{e}^\top\right| = v$.

We say that the variable nodes corresponding to the support of such an error pattern constitute a (u, v) trapping set as defined in [Ric03].

Let us use the polynomial representation of a circulant block

$$\mathrm{h} = \sum_{i \in \mathrm{Supp(h)}} x^i \in \mathbb{F}_2[x]/(x^r - 1)\,.$$

Applying the Frobenius endomorphism we obtain:

$$\mathrm{h}^2 = \sum_{i \in \mathrm{Supp(h)}} x^{2i}\,.$$

Since $\forall i \in \{0,1\}, \left|\mathrm{h}_i^2\right| = \left|\mathrm{h}_i\right| = d$, we have identified (d,d) near-codewords present in any QC-MDPC code.

In the following definition, we describe all the (d,d) near-codewords based on this template, *i.e.* those of each circulant block and all their circular shifts.

Definition 27. Let $\mathrm{h} = (\mathrm{h}_0 \quad \mathrm{h}_1) \in \mathbb{F}_2[x]/(x^r-1)^2$ be the parity check matrix of a QC-MDPC code of row weight w.

We write $\mathcal{N}$ the following set of (d,d) near-codewords $\mathrm{e} = (\mathrm{e}_0, \mathrm{e}_1)$

$$\mathcal{N} = \{(x^s \mathrm{h}_0, 0) \,|\, s \in \{0, \dots, r-1\}\} \cup \{(0, x^s \mathrm{h}_1) \,|\, s \in \{0, \dots, r-1\}\}\ .$$

There exists $2r$ such (d,d) near-codewords.

We will also consider the set $2\mathcal{N}$ of the sums of two (d,d) near-codewords as they are $(2d - \epsilon_0, 2d - \epsilon_1)$ near-codewords (for some small values ϵ_0 and ϵ_1) and we usually have $2d = w \approx t$

5.2 Error patterns impeding decoding

We have already covered the case where one uses a minimum distance decoder in a QC-MDPC scheme. But such a decoder is too challenging to efficiently implement, and we use a bit-flipping or a belief propagation algorithm. We will here focus on the BGF variant of the former [DGK20].

In any bit-flipping variant, the syndrome is used to compute the counters that are then used to decide whether to flip a bit or not. In this subsection we will use those two data as a benchmark to assess the influence of low weight codewords and near-codewords on the decoder.

We keep the same construction as before and choose an error pattern e' as

$$\mathrm{e} = \mathrm{c} + \mathrm{e}'$$

with c in $\mathcal{C}$, $\mathcal{N}$ or $2\mathcal{N}$ and $|\mathrm{e}| = t$, but this time we relax the condition on the weight of e'. We have

$$|\mathrm{e}'| = |\mathrm{c}| + |\mathrm{e}| - 2\,|\mathrm{e} \star \mathrm{c}|\ .$$

For a set $\mathcal{S}$ that is either $\mathcal{C}$, $\mathcal{N}$ or $2\mathcal{N}$, we define the set of error patterns e that are near $\mathcal{S}$. We set the distance to these sets via the constant $\ell = |\mathrm{e} \star \mathrm{c}|$.

Definition 28. Let $S \subset (\mathbb{F}_2[x]/(x^r-1))^2$, we write

$$\mathcal{A}_{t,\ell}(S) = \bigcup_{v \in S} \{u \in (\mathbb{F}_2[x]/(x^r-1))^2 \ |\ |u| = t, |u \star v| = \ell\}\,.$$

Algorithm 5 provides a way to construct these sets.

These sets can also be seen as unions of spheres of radius $w + t - 2\ell$ and centers in $\mathcal{S}$.

Algorithm 5: Weak error patterns generation.

```
function weak_error($r, t, \ell, (\mathrm{h}_0, \mathrm{h}_1), S$):
    input  : Block size $r$, error weight $t$, an integer $\ell$,
             a key $(\mathrm{h}_0, \mathrm{h}_1) \in (\mathbb{F}_2[x]/(x^r - 1))^2$, a set $S \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$.
    output : A vector c that has $\ell$ intersections with an element of $S$.
    $\mathrm{c} \leftarrow$ sample$_S(\mathrm{h}_0, \mathrm{h}_1)$;
    $s \overset{\$}{\leftarrow} \{0, \dots, r-1\}$;
    $(p_0, \dots, p_{\ell-1}) \overset{\$}{\leftarrow}$ Sample $\ell$ values from Supp(c);
    $(p_\ell, \dots, p_{t-1}) \overset{\$}{\leftarrow}$ Sample $(t-\ell)$ values from $\{0, \dots, r-1\} \setminus$ Supp(c);
    $\mathrm{c} \leftarrow 0$;
    for $k \in \{0, \dots, t-1\}$ do
        $c_{s+p_k} \leftarrow 1$;
    return c;

function sample$_\mathcal{C}(\mathrm{h}_0, \mathrm{h}_1)$:
    return $(\mathrm{h}_1, \mathrm{h}_0)$;

function sample$_\mathcal{N}(\mathrm{h}_0, \mathrm{h}_1)$:
    $(\mathrm{c}_0, \mathrm{c}_1) \leftarrow (0, 0)$;
    $i \overset{\$}{\leftarrow} \{0, 1\}$;
    $\mathrm{c}_i \leftarrow \mathrm{h}_i$;
    return $(\mathrm{c}_0, \mathrm{c}_1)$;

function sample$_{2\mathcal{N}}(\mathrm{h}_0, \mathrm{h}_1)$:
    $s \overset{\$}{\leftarrow} \{0, \dots, r-1\}$;
    return sample$_\mathcal{N}(\mathrm{h}_0, \mathrm{h}_1) + x^s$sample$_\mathcal{N}(\mathrm{h}_0, \mathrm{h}_1)$;
```

Proposition 29. *Suppose that there exists a weight w such that $\forall u \in S, |u| = w$, then $\forall v \in \mathcal{A}_{t,\ell}(S), |v - u| = w + t - 2\ell$ and*

$$\left|\mathcal{A}_{t,\ell}(S)\right| \leq |S| \binom{w}{\ell} \binom{n-w}{t-\ell} .$$

In this case, we write

$$\mathcal{D}_{t,w+t-2\ell}(S) = |S| \frac{\binom{w}{\ell}\binom{n-w}{t-\ell}}{\binom{n}{t}} ,$$

an upper bound on the density of the set $\mathcal{A}_{t,\ell}(S)$.

Influence on decoding. Let $\mathcal{S}$ be $\mathcal{C}$, $\mathcal{N}$ or $2\mathcal{N}$. The proximity of an error pattern to an element of $\mathcal{S}$ has an influence on the counters. Let us write one of these elements $\mathrm{c} \in \mathcal{S} \in (\mathbb{F}_2[x]/(x^r - 1))^2$ and an error pattern $\mathrm{e} \in (\mathbb{F}_2[x]/(x^r - 1))^2$. Let us separate the positions in c and e depending on whether they are mutual or not.

We can write

$$\mathrm{p} = \mathrm{e} \star \mathrm{c} , \qquad \mathrm{e}^\perp = \mathrm{e} - \mathrm{p} , \qquad \mathrm{c}^\perp = \mathrm{c} - \mathrm{p} ,$$

then

$$\mathrm{e} = \mathrm{e}^{\perp} + \mathrm{p}\,, \qquad \mathrm{c} = \mathrm{c}^{\perp} + \mathrm{p}\,.$$

Since the distance between e and c is given by the formula

$$|\mathrm{e}| + |\mathrm{c}| - 2\,|\mathrm{p}|\ ,$$

and assuming the elements of $\mathcal{S}$ all have the same weight, a closest element of $\mathcal{S}$ to e is one that maximizes $\ell = |\mathrm{p}|$.

We now suppose that c is one of those. In Table 3, we give statistics for different sets $\mathcal{S}$ and different values of ℓ, namely the syndrome weight and the distribution of the counters classified according to whether they concern a position belonging to $\mathrm{e}^{\perp}$, $\mathrm{h}^{\perp}$, p or none of these. Remember that for a bit-flipping iteration to be efficient, the counters of positions in the support of e have to be higher than the others. What we can observe from the table is that as ℓ increases, the counters of positions in the support of $\mathrm{e}^{\perp}$ or $\mathrm{c}^{\perp}$ increase, while those of p and the others decrease. In other words, the algorithm is more likely to mistakenly add errors by flipping positions in the support of $\mathrm{c}^{\perp}$ and miss the errors in $\mathrm{p} \subset \mathrm{e}$. It is also interesting to point out that this influence on the counters as well as on the syndrome weight can even be observed for small values of ℓ.

Note that as all the sets $\mathcal{C}$, $\mathcal{N}$ or $2\mathcal{N}$ are stable by blockwise circular shifts, any error pattern has at least one intersection with one of them, and is highly likely to have two. Therefore, depending on the set $\mathcal{S}$, the set $\mathcal{A}_{t,\ell}(\mathcal{S})$ might be empty for lower values of ℓ.

In Figure 8, we show the DFR curves for QC-MDPC codes (drawn uniformly at random) with parameters $(d, t) = (21, 42)$ and varying block size r on the x-axis. The solid line curve is obtained with uniformly randomly drawn error patterns and the others are with sets of error patterns close to the near-codewords. Note that here we measure the contribution of each set of error patterns to the average DFR; the measured DFR is therefore multiplied by the density of the sets. With these *toy* parameters, we can already see, using simulations, an error floor for random error patterns. Moreover, it appears that with the near-codewords, we have indeed identified the main contributor to the error floor.

5.3 A good approximation of the density of the error patterns of interest

In this subsection, we show that by considering the union bound to obtain the density of error patterns close to the near-codewords, we do not overestimate. In other words, we show here that the spheres whose union we take do not intersect too much. This is in the case of a key drawn uniformly at random, *i.e.* which is not a weak key.

There exists r near-codewords for each block. Let us fix an error pattern e and, for $k \in \{1, \dots, 2r\}$ and $i \in \{0, \dots, d\}$, write the events $N_{i,k}$:

$$\forall k \in \{0, \dots, r-1\}, N_{i,k} := \{|(x^s \mathrm{h}_0, 0) \star \mathrm{e}| = i\}\,, N_{i,k+r} := \{|(0, x^s \mathrm{h}_1) \star \mathrm{e}| = i\}\,.$$

The density of the error patterns with exactly i intersections with at least

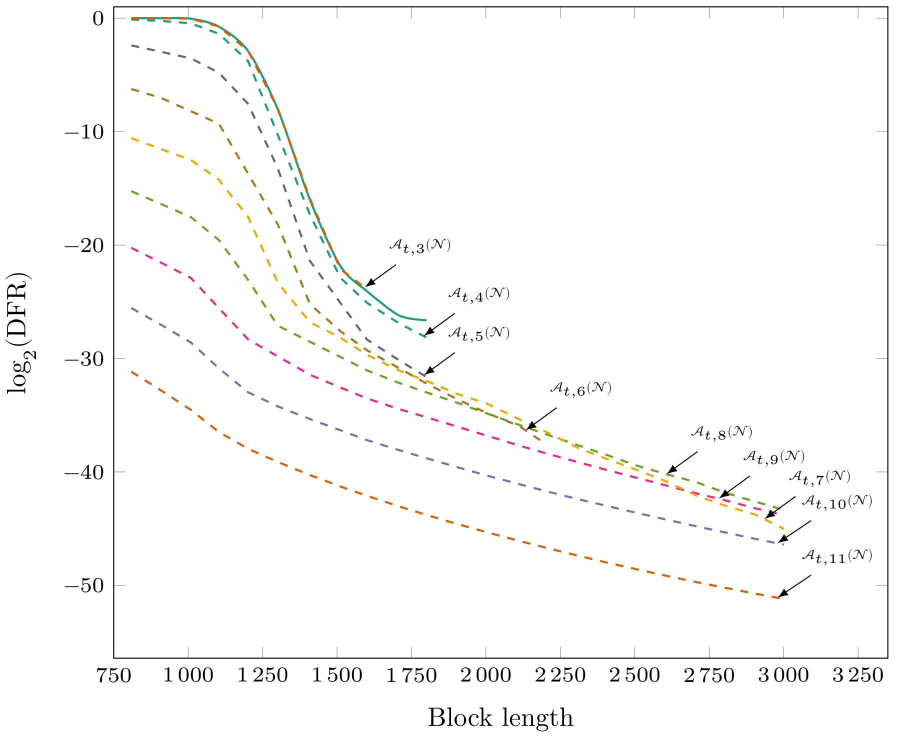

Figure 8: Toy example. Solid line: error patterns taken uniformly at random. Dashed lines: DFR with density for error pattern close to near-codewords. $(d, t) = (21, 42)$.

one near-codeword is then

$$\Pr\left[\bigcup_{k=0}^{2r-1} N_{i,k}\right].$$

We write $\rho_i = \frac{\sum_{0\le k<\ell<2r}\Pr(N_{i,k}\cap N_{i,\ell})}{\sum_{0\le k<2r}\Pr(N_{i,k})}$, by Bonferroni inequalities, we know that

$$(1-\rho_i)\sum_{0\le k<2r}\Pr\left(N_{i,k}\right) \le \Pr\left[\bigcup_{k=0}^{2r-1} N_{i,k}\right] \le \sum_{0\le k<2r}\Pr\left(N_{i,k}\right).$$

Let us now derive an upper bound for ρ_i in the case where the parity check matrix $\mathrm{h} = (\mathrm{h}_0, \mathrm{h}_1)$ is drawn uniformly at random. Using Corollary 41 in §A, we know the probability of two rows[4] having I intersections in a circulant block chosen uniformly at random, hence, for two near-codewords:

$$\forall j \in \{0,1\}, \forall s \neq t, \quad p_I := \Pr\left[\left|x^s\mathrm{h}_j \star x^t\mathrm{h}_j\right| = I\right] = \frac{r}{d-I}\frac{\binom{d-1}{d-I-1}\binom{r-d-1}{d-I-1}}{\binom{r}{d}}.$$

Thus, using the usual techniques for calculating the probability that the error vector of weight t has exactly i intersections with two near-codewords of weight d and intersecting each other I times, we have:

$$\sum_{0\le k<\ell<2r}\Pr\left(N_{i,k}\cap N_{i,\ell}\right) = \sum_{\substack{0\le k<r\\ r\le \ell<2r}}\Pr\left(N_{i,k}\cap N_{i,\ell}\right) + \sum_{\substack{0\le k<\ell<r\\ \text{or}\\ r\le k<\ell<2r}}\Pr\left(N_{i,k}\cap N_{i,\ell}\right)$$
$$\le \frac{r^2}{\binom{n}{2}}\frac{\binom{d}{i}^2\binom{n-2d}{t-2i}}{\binom{n}{t}} + \frac{2\binom{r}{2}}{\binom{n}{2}}\sum_{0\le I\le d} p_I \sum_{0\le a\le i}\frac{\binom{d-I}{a}^2\binom{I}{i-a}\binom{n-2d+I}{t-i-a}}{\binom{n}{t}}.$$

We give in Table 4 a few values of ρ_i for BIKE parameters. We can see that apart from small values of i (*i.e.* less than 6 for BIKE parameters), the gap between the two bounds is small enough to disappear when we take the logarithm.

In Figure 9 we give the density of all the sets of error patterns described in the previous subsection for $(r,d,t) = (12\,323, 71, 134)$. We can see, for each set, the range of distance of interest for our analysis *i.e.* those that have a density greater that 2^{-128}.

5.4 Lower bound on the DFR with simulations

Using Algorithm 5, we can evaluate the DFR on the sets $\mathcal{A}_{t,\ell}(\mathcal{S})$ for any decoding algorithm, any ℓ, and $\mathcal{S} \in \{\mathcal{C}, \mathcal{N}, 2\mathcal{N}\}$. In §B, we give DFR estimates for the BGF decoder. We can observe that the product of the density of the sets $\mathcal{A}_{t,\ell}(\mathcal{C})$, $\mathcal{A}_{t,\ell}(\mathcal{N})$, $\mathcal{A}_{t,\ell}(2\mathcal{N})$ multiplied by the obtained DFR is increasing. But we eventually reach the limits of what can be obtained with simulation. It would not be surprising if as the distance increases, the DFR converges to its average value (which is expected to be below $2^{-\lambda}$).

We can also see in those figures that the near-codewords in $\mathcal{N}$ seem to have the most influence on the decoder in comparison to the other sets $\mathcal{C}$ or $2\mathcal{N}$.

[4] Remember that, in a circulant matrix, there is a one-to-one mapping between rows and columns that simply revert the vectors.

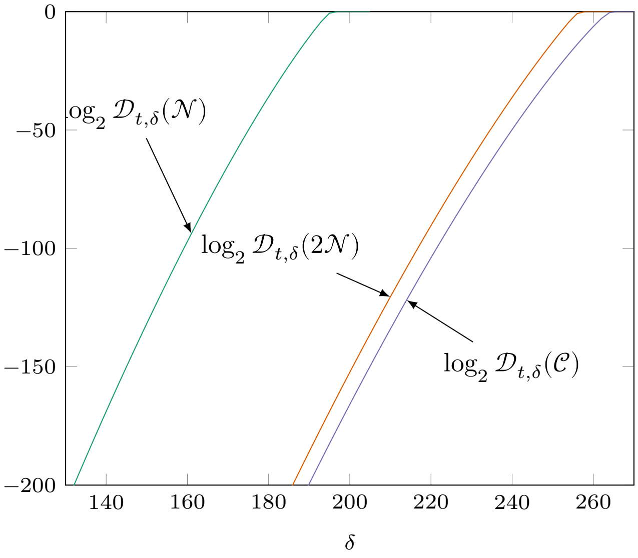

Figure 9: Upper bound on the probability that an error pattern of weight t is at a certain distance δ from $\mathcal{N}$, $2\mathcal{N}$ or $\mathcal{C}$.
For $\ell = |\mathrm{e} \star \mathrm{c}|$, the distance is $\delta = |\mathrm{c}| + t - 2\ell$ where $|\mathrm{c}|$ is d, w or w respectively for $\mathcal{N}$, $2\mathcal{N}$ or $\mathcal{C}$.
$(r, d, t) = (12\,323, 71, 134)$.

5.5 Comments

Note that the DFR obtained are not the results of extrapolations and this construction gives directly a lower bound on the DFR. The (small) imprecision on these values comes from the fact that we only compute a bound on the density of the considered sets and the fact that the DFR is estimated with simulation. As discussed in the previous subsection, the former is really tight and, for the latter, estimations are given with their 99% confidence interval.

While decoders are slowed down by the closeness of problematic patterns, we observe that failures would not be avoided by simply increasing the number of iterations. Error patterns are either decoded in a few iterations or they are never decoded.

6 Conclusion

With this document, we have shown that we now have a better comprehension of the DFR for QC-MDPC decoders. We rely on an extrapolation method for which we have shown how to rigorously assess the precision. We have highlighted specific patterns that can affect decoding performance. Indeed, we have studied both sides of the product used to compute the syndrome: specific parity check matrices (giving weak keys) and specific error patterns (giving the error floor). In a way, provided the decoding assumption is true, the extrapolation method gives an upper bound on the DFR while the weak keys and error floor work gives a lower bound. Both methods are eventually limited by the amount of computational power required to perform extensive simulations. However, at this point, they appear consistent with each other. These results reinforce our belief that the BIKE parameters, as they are today, do indeed meet the security requirements related to the DFR.

References

[APRS20] Daniel Apon, Ray A. Perlner, Angela Robinson, and Paolo Santini. "Cryptanalysis of LEDAcrypt". In: *Advances in Cryptology - CRYPTO*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, 2020, pp. 389–418. DOI: 10.1007/978-3-030-56877-1_14.

[AYU20] Nuh Aydin, Bahattin Yildiz, and Suleyman Uludag. "A Class of Weak Keys for the QC-MDPC Cryptosystem". In: *Algebraic and Combinatorial Coding Theory (ACCT)*. IEEE, 2020, pp. 1–4. DOI: 10.1109/ACCT51235.2020.9383383.

[BDLO16] Magali Bardet, Vlad Dragoi, Jean-Gabriel Luque, and Ayoub Otmani. "Weak Keys for the Quasi-Cyclic MDPC Public Key Encryption Scheme". In: *AFRICACRYPT 2016*. Ed. by David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi. Vol. 9646. LNCS. Springer, 2016, pp. 346–367. DOI: 10.1007/978-3-319-31517-1_18.

[BIKE] Carlos Aguilar Melchor, Nicolas Aragon, Paulo S L M Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Ghosh Santosh, Shay Gueron, Tim Güneysu, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. *BIKE*. NIST Round 3 submission for Post-Quantum Cryptography. Aug. 2020. URL: https://bikesuite.org.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. "Decoding Random Binary Linear Codes in $2^{n/20}$: How $1+1=0$ Improves Information Set Decoding". In: *Advances in Cryptology - EUROCRYPT*. LNCS. Springer, 2012. DOI: 10.1007/978-3-642-29011-4_31.

[BMT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. "On the inherent intractability of certain coding problems". In: *IEEE Transactions on Information Theory* 24.3 (May 1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[BT02] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to probability*. English. Belmont, Mass.: Athena Scientific, 2002. ISBN: 978-1886529403.

[CP34] Charles J Clopper and Egon S Pearson. "The use of confidence or fiducial limits illustrated in the case of the binomial". In: *Biometrika* 26.4 (Dec. 1934), pp. 404–413. ISSN: 0006-3444. DOI: 10.1093/biomet/26.4.404.

[CS15] Rodolfo Canto Torres and Nicolas Sendrier. “Analysis of Information Set Decoding for a Sub-linear Error Weight”. In: *Post-Quantum Cryptography (PQCrypto)*. Ed. by Tsuyoshi Takagi. 2015, pp. 144–161. DOI: 10.1007/978-3-319-29360-8_10,.

[Den03] Alexander W Dent. “A Designer’s Guide to KEMs”. In: *9th IMA International Conference on Cryptography and Coding*. Ed. by Kenneth G. Paterson. Vol. 2898. LNCS. Cirencester, UK: Springer, Dec. 2003, pp. 133–151. DOI: 10.1007/978-3-540-40974-8_12.

[DGK19] Nir Drucker, Shay Gueron, and Dusan Kostic. *On constant-time QC-MDPC decoding with negligible failure rate*. Cryptology ePrint Archive, Report 2019/1289. 2019. URL: https://eprint.iacr.org/2019/1289.

[DGK20] Nir Drucker, Shay Gueron, and Dusan Kostic. “QC-MDPC Decoders with Several Shades of Gray”. In: *Post-Quantum Cryptography (PQCrypto)*. Ed. by Jintai Ding and Jean-Pierre Tillich. Vol. 12100. LNCS. Springer, 2020, pp. 35–50. DOI: 10.1007/978-3-030-44223-1_3.

[Dum91] Ilya Dumer. “On minimum distance decoding of linear codes”. In: *Fifth Joint Soviet–Swedish International Workshop on Information Theory*. Ed. by Grigori A Kabatianskii. Moscow, Russia, 1991, pp. 50–52.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. In: *CRYPTO’99*. Ed. by Michael J. Wiener. Vol. 1666. LNCS. Santa Barbara, CA, USA: Springer, Aug. 1999, pp. 537–554. DOI: 10.1007/3-540-48405-1_34.

[Gar+01] Roberto Garello, Franco Chiaraluce, Paola Pierleoni, Marco Scaloni, and Sergio Benedetto. “On error floor and free distance of turbo codes”. In: *IEEE International Conference on Communications (ICC)*. Vol. 1. IEEE, 2001, 45–49 vol.1. DOI: 10.1109/ICC.2001.936270.

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. “A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors”. In: *Advances in Cryptology - ASIACRYPT*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. 2016, pp. 789–815. ISBN: 978-3-662-53886-9. DOI: 10.1007/978-3-662-53887-6_29.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A modular analysis of the Fujisaki-Okamoto transformation”. In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371. DOI: 10.1007/978-3-319-70500-2_12.

[McE78] Robert J. McEliece. “A Public-Key System Based on Algebraic Coding Theory”. In: DSN Progress Report 42-44. Jet Propulsion Lab, 1978, pp. 114–116. URL: https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. “Decoding random linear codes in $O(2^{0.054n})$”. In: *Advances in Cryptology - ASIACRYPT*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, 2011, pp. 107–124. DOI: 10.1007/978-3-642-25385-0_6.

[MO15] Alexander May and Ilya Ozerov. “On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes”. In: *Advances in Cryptology - EUROCRYPT*. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 203–228. DOI: 10.1007/978-3-662-46800-5_9.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes”. In: *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2013, pp. 2069–2073. DOI: 10.1109/ISIT.2013.6620590.

[Pra62] Eugene Prange. “The use of information sets in decoding cyclic codes”. In: *IRE Transactions on Information Theory* 8.5 (Sept. 1962), pp. 5–9. DOI: 10.1109/TIT.1962.1057777.

[Ric03] Tom Richardson. “Error Floors of LDPC Codes”. In: *41st Annual Allerton Conference on Communication, Control, and Computing*. 2003, pp. 1426–1435.

[Sen11] Nicolas Sendrier. “Decoding One Out of Many”. In: *Post-Quantum Cryptography (PQCrypto)*. Vol. 7071. LNCS. 2011, pp. 51–67. DOI: 10.1007/978-3-642-25405-5_4.

[Spr79] Melvin Dale. Springer. *The Algebra of random variables*. English. New York: Wiley, 1979. ISBN: 978-0471014065.

[Ste88] Jacques Stern. “A method for finding codewords of small weight”. In: *Coding Theory and Applications*. Ed. by Gérard Cohen and Jacques Wolfmann. Vol. 388. LNCS. Springer, 1988, pp. 106–113. DOI: 10.1007/BFb0019850.

[SV19] Nicolas Sendrier and Valentin Vasseur. “On the Decoding Failure Rate of QC-MDPC Bit-Flipping Decoders”. In: *Post-Quantum Cryptography (PQCrypto)*. Ed. by Jintai Ding and Rainer Steinwandt. Vol. 11505. LNCS. Chongqing, China: Springer, May 2019, pp. 404–416. DOI: 10.1007/978-3-030-25510-7_22.

[SV20] Nicolas Sendrier and Valentin Vasseur. *On the existence of weak keys for QCMDPC decoding*. Cryptology ePrint Archive, Report 2020/1232. 2020. URL: https://eprint.iacr.org/2020/1232.

[Til18] Jean-Pierre Tillich. “The Decoding Failure Probability of MDPC Codes”. In: *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 941–945. DOI: 10.1109/ISIT.2018.8437843.

[Vas21] Valentin Vasseur. “Post-quantum cryptography: a study of the decoding of QC-MDPC codes”. PhD Thesis. Université de Paris, Mar. 2021. URL: https://tel.archives-ouvertes.fr/tel-03254461.

A New properties on the distance spectrum

In [GJS16] the notion of spectrum for a circulant matrix was introduced. We recall its definition and significance here and show its close relationship with the intersections between two columns in the same circulant block. We will see that due to the quasi-cyclic nature of the parity check matrix, the probability that two columns intersect in at least one row is lower than with independent random vectors of the same fixed weight (for BIKE parameters).

Definition 30. We define the distance between two positions $i, j \in \{0, \dots, r-1\}$ in a circulant block as

$$\mathrm{d}(i,j) = \min((r+j-i) \bmod r, (r+i-j) \bmod r) \,.$$

The *spectrum* of h is defined as the multiset of all distances, their multiplicity being given by μ:

$$\mathrm{Sp}(\mathrm{h}) = \{(\delta, \mu(\delta, \mathrm{h})) \mid \delta \in \{0, 1, \dots, \lfloor r/2 \rfloor\}\} \,.$$

The *spectral polynomial* of h is defined as:

$$\mathrm{s}(\mathrm{h}) = \sum_{\delta=1}^{\lfloor r/2 \rfloor} \mu(\delta, \mathrm{h})\, x^{\delta} \,.$$

It is shown in [GJS16] that the knowledge, even partial, of the distance spectrum of a sparse polynomial allows its complete recovery. It is also shown that the statistical analysis of error patterns leading to failures of the QC-MDPC decoder provides information on the secret key spectrum and eventually allows a key recovery attack.

New properties of the distance spectrum.

Proposition 31. *Let* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$. *We write, for any* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$, $\mathrm{h}^\top = \phi_{-1}(\mathrm{h}) = \sum_{i \in \mathrm{Supp}(\mathrm{h})} x^{-i}$. *This operation corresponds to transposing the circulant block. We have*

$$\mathrm{h}\mathrm{h}^\top = |\mathrm{h}| + \mathrm{s}(\mathrm{h}) + \mathrm{s}(\mathrm{h})^\top$$

where the above product of polynomials is considered in $\mathbb{Z}[x]/(x^r - 1)$.

Proof. By expanding the product then decomposing the sum, we obtain

$$\begin{aligned}
\mathrm{h}\mathrm{h}^\top &= \left(\sum_{i \in \mathrm{Supp}(\mathrm{h})} x^i\right)\left(\sum_{j \in \mathrm{Supp}(\mathrm{h})} x^{-j}\right) \\
&= \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i=j}} x^{i-j} + \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i>j}} x^{i-j} + \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i<j}} x^{i-j} \\
&= \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i=j}} x^{i-j} + \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i<j}} x^{\min(i-j, r+i-j)} + \sum_{\substack{i,j \in \mathrm{Supp}(\mathrm{h}) \\ i<j}} x^{\max(i-j, r+i-j)} \\
&= |\mathrm{h}| + \mathrm{s}(\mathrm{h}) + \mathrm{s}(\mathrm{h})^\top \,.
\end{aligned}$$

□

There is a one-to-one correspondence between the spectrum of a polynomial h and the number of common bits between h and its δ-shift x^δh.

Corollary 32. *For all* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ *and all* $\delta \in \{0, 1, \dots, \lfloor r/2 \rfloor\}$*, we have*

(i) $\mu(\delta, \mathrm{h}) = \left|\mathrm{h} \star x^\delta \mathrm{h}\right|$*;*

(ii) if $\delta \neq 0$*,* $\mu(1, \mathrm{h}) = \mu(\delta, \phi_\delta(\mathrm{h}))$*.*

Proof. Considering coefficients indices modulo r:

$$\mu(\delta, \mathrm{h}) = (\mathrm{h}\mathrm{h}^\top)_\delta = \sum_{i=0}^{r-1} h_i h_{i-\delta} = \left|\mathrm{h} \star x^\delta \mathrm{h}\right| .$$

The second identity easily derives from $\phi_\delta(x\mathrm{h}) = x^\delta \phi_\delta(\mathrm{h})$ and the fact that ϕ_δ is an isometry for the Hamming distance. □

Remark 33. $\mathrm{Sp}(\mathrm{h}) = \mathrm{Sp}(x^\ell \mathrm{h})$ for all $\ell \in \{0, \dots, r-1\}$.

Proposition 34. *The application*

$$\begin{aligned} \mathrm{Sp}(\mathrm{h}) &\to \mathrm{Sp}(\phi_a(\mathrm{h})) \\ (\delta, m) &\mapsto (\delta', m) \end{aligned}$$

with $\delta' = \min(a\delta \bmod r, r - (a\delta \bmod r))$*, is a bijection.*

Proof. $\phi_\delta(\mathrm{h}\mathrm{h}^\top) = \phi_\delta(\mathrm{h})\phi_\delta(\mathrm{h})^\top$ □

Distance spectrum statistics.

Definition 35. Let $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$. Viewed as a binary vector, we assume h starts with 0 and ends with 1, using run-length encoding it can be uniquely described as a sequence of strictly positive integers $(z_1, o_1, z_2, o_2, \dots, z_s, o_s)$: it starts with z_1 zeros followed by o_1 ones, followed by z_2 zeros, *etc.*

We will write $\mathrm{h} \sim (z_1, o_1, z_2, o_2, \dots, z_s, o_s)$.

Proposition 36. *If* $\mathrm{h} \sim (z_1, o_1, z_2, o_2, \dots, z_s, o_s)$*, then*

$$\mu(1, \mathrm{h}) = \sum_{i=1}^{s} (o_i - 1) .$$

Proof. Counting the multiplicity of $\delta = 1$ in h is simply counting the number of times there are two consecutive ones. In a block of o_i consecutive ones, this happens $(o_i - 1)$ times. □

The following proposition reduces the problem of fixing a multiplicity in a vector to splitting $|\mathrm{h}|$ ones into s nonempty segments interleaved with s nonempty segments of zeros of total length $(r - |\mathrm{h}|)$.

Proposition 37. *Let* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$*. Suppose* $\mathrm{h} \sim (z_1, o_1, z_2, o_2, \dots, z_s, o_s)$*, then*

$$\begin{cases} o_1 + o_2 + \cdots + o_s = |\mathrm{h}| \ ; \\ z_1 + z_2 + \cdots + z_s = r - |\mathrm{h}| \end{cases}$$

and

$$\mu(1, \mathrm{h}) = m \quad \textit{if and only if} \quad s = |\mathrm{h}| - m .$$

Corollary 38. *The number of polynomials* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ *of weight* d *starting with a zero and ending with a one such that* $\mu(1, \mathrm{h}) = m$ *is exactly*

$$\binom{d-1}{d-m-1}\binom{r-d-1}{d-m-1}.$$

Proof. The equivalence of the previous proposition gives $s = d - m$, and patterns following the two other conditions are counted using the "stars and bars" principle. □

To count the number of general patterns h such that $\mu(1, \mathrm{h}) = m$, circular shifts of patterns starting with a zero and ending with a one have to be considered. However not all shifts are possible as we need to avoid counting several times the same configuration. For example shifting $\mathrm{h} \sim (z_1, o_1, z_2, o_2, \dots, z_s, o_s)$ by $(z_1 + o_1)$ positions would give $x^{-(z_1+o_1)}\mathrm{h} \sim (z_2, o_2, \dots, z_s, o_s, z_1, o_1)$.

For the sake of clarity, let us generalize our $\sim$ notation for any pattern h.

Definition 39. Let $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ and let ℓ be the smallest integer such that $x^{-\ell}\mathrm{h}$ starts with a 0 and ends with a 1. We write

$$\mathrm{h} \sim (z_1, o_1, \dots, z_s, o_s)^{\ell}$$

if and only if

$$x^{-\ell}\mathrm{h} \sim (z_1, o_1, \dots, z_s, o_s).$$

Proposition 40. *For any polynomial* $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ *of weight* d *such that* $\mu(1, \mathrm{h}) = m$*, there is a unique representation*

$$\mathrm{h} \sim (z_1, o_1, \dots, z_s, o_s)^{\ell} \quad \textit{with} \quad \begin{cases} s = d - m\,; \\ o_1 + \dots + o_s = d\,; \\ z_1 + \dots + z_s = r - d\,; \\ \ell \in \{0, \dots, z_1 + o_1 - 1\}\,. \end{cases}$$

Corollary 41. *For given integers* m*,* $0 \le m < d$*, and* δ*,* $1 \le \delta \le \lfloor r/2 \rfloor$*, there are*

$$\mathcal{N}_m := \frac{r}{d-m}\binom{d-1}{d-m-1}\binom{r-d-1}{d-m-1}$$

polynomials $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ *of weight* d *such that* $\mu(\delta, \mathrm{h}) = m$.

Proof. When $\delta = 1$ the result derives from Proposition 40.

- First, if $m < d - 1$, all values $z_1, \dots, z_s$ and $o_1, \dots, o_s$ are at least 1, so $z_1 \in \{1, \dots, r - d - (s-1)\}$ and $o_1 \in \{1, \dots, d - (s-1)\}$. If z_1 and o_1 are fixed, Corollary 38 tells us that there are $\binom{d-o_1-1}{d-m-2}\binom{r-d-z_1-1}{d-m-2}$ ways to choose the values of $z_2, \dots, z_s$ and $o_2, \dots, o_s$. Accounting for the different $(z_1 + o_1)$ possible values of ℓ, we obtain

$$\sum_{z_1=1}^{r-2d+m+1}\sum_{o_1=1}^{m+1}(z_1 + o_1)\binom{d-o_1-1}{d-m-2}\binom{r-d-z_1-1}{d-m-2}$$
$$= \frac{r}{d-m}\binom{d-1}{d-m-1}\binom{r-d-1}{d-m-1}.$$

- Now if $m = d - 1$, then the only relevant patterns are the r shifts of the pattern consisting of z_1 consecutive zeros followed by o_1 ones.

The generalization to any value of δ derives from the identity $\mu(1, \mathrm{h}) = \mu(\delta, \phi_\delta(\mathrm{h}))$ of Corollary 32. □

Remark 42. The probability that two binary vectors of length r and weight d drawn uniformly at random have m intersections is

$$\frac{\binom{d}{m}\binom{r-d}{d-m}}{\binom{r}{d}} = \frac{d}{d-m}\frac{r-d}{r}\frac{\mathcal{N}_m}{\binom{r}{d}} .$$

Thus, in a quasi-cyclic matrix, for $m > \left\lfloor \frac{d^2}{r} \right\rfloor$, a column is less likely to have exactly m of intersections with another column in its own circulant block than with a column from other blocks. In fact, with BIKE parameters, $\left\lfloor \frac{d^2}{r} \right\rfloor = 0$.

Corollary 43. *Let r and d be positive integers and let $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$ be a pattern of weight d. For any integer δ such that $1 \leq \delta \leq \lfloor r/2 \rfloor$, the probability that δ has a nonzero mutiplicity in the spectrum of* h *is*

$$1 - \frac{\mathcal{N}_0}{\binom{r}{d}} = 1 - \frac{\binom{r-d-1}{d-1}}{\binom{r-1}{d-1}} .$$

Corollary 44. *We assume the independence of the multiplicities of the spectrum. Let m be an integer such that $0 \leq m < d$, δ be such that $1 \leq \delta \leq \lfloor r/2 \rfloor$ and $\mathrm{h} \in \mathbb{F}_2[x]/(x^r - 1)$,*

$$\pi_m = \Pr[\mu(\delta, \mathrm{h}) = m] = \frac{\mathcal{N}_m}{\binom{r}{d}} ,$$

$$p_{\geq m} = \Pr\left[\max_{\delta \in \{1,\dots,\lfloor r/2 \rfloor\}} \mu(\delta, \mathrm{h}) \geq m\right] = 1 - (1 - \pi_m)^{\lfloor r/2 \rfloor} ,$$

and

$$p_{=m} = \Pr\left[\max_{\delta \in \{1,\dots,\lfloor r/2 \rfloor\}} \mu(\delta, \mathrm{h}) = m\right] = p_{\geq m} - p_{\geq m+1} .$$

Example. We give in Table 5, for $(r, d) = (12\,323, 71)$, the probabilities of having a certain multiplicy in the spectrum of a circulant block. We observe it is typically low. Note that the independence of the multiplicities for all distances is not true in general. However, this approximation is really close to the values observed in simulation.

Reconstructing the secret key from the spectrum. Proposition 31 brings a new approach to the problem of reconstructing a key from its spectrum used in [GJS16]. Knowing Sp(h), one can write the following system of equations in r boolean variables $h_0, \dots, h_{r-1} \in \{0, 1\}$.

$$\begin{cases} d = \mu(0, \mathrm{h}) = h_0^2 + \dots + h_{r-1}^2 = h_0 + \dots + h_{r-1} \\ \mu(1, \mathrm{h}) = h_0 h_1 + h_1 h_2 + \dots + h_{r-1} h_0 \\ \vdots = \vdots \\ \mu(\lfloor (r-1)/2 \rfloor, \mathrm{h}) = h_0 h_{\lfloor \frac{r-1}{2} \rfloor} + h_1 h_{\lfloor \frac{r-1}{2} \rfloor + 1} + \dots + h_{r-1} h_{\lfloor \frac{r-1}{2} \rfloor - 1} \end{cases}$$

Such a system is said to be *pseudo-boolean* as it involves linear operations in $\mathbb{Z}$ of boolean values. Most satisfiability modulo theories (SMT) solvers implement a way to solve them. Nevertheless, the resolution of this system can greatly benefit from simplification.

Indeed, say, for some δ, $\mu(\delta, \mathrm{h}) = 0$. Then

$$h_0 h_\delta + h_1 h_{\delta+1} + \cdots + h_{r-1} h_{\delta-1} = 0$$

can be written as the conjunctive normal form (CNF)

$$\overline{h_0 h_\delta} \wedge \overline{h_1 h_{\delta+1}} \wedge \cdots \wedge \overline{h_{r-1} h_{\delta-1}} \equiv \left(\overline{h_0} \vee \overline{h_\delta}\right) \wedge \left(\overline{h_1} \vee \overline{h_{\delta+1}}\right) \wedge \cdots \wedge \left(\overline{h_{r-1}} \vee \overline{h_{\delta-1}}\right).$$

To further simplify the system, one can use the fact that at least one multiplicity is nonzero and a lot of variables can be removed. This comes from the fact that the equation system has at least $2r$ solutions: if one vector is a solution then so are its circular shifts and the transpositions of its circular shifts. Therefore, if we know that $\mu(\delta', \mathrm{h}) > 0$ then we can fix $h_0 = h_{\delta'} = 1$ and the above CNF gives $h_\delta = h_{r-\delta} = h_{\delta+\delta'} = h_{\delta'-\delta} = 0$. So each zero multiplicity in the spectrum can fix the value of at most four variables.

Reconstructing a block with $r = 12\,323$ and $d = 71$ from this simplified system takes less than a second using the SMT solver z3. In comparison, the algorithm proposed in [GJS16] is a depth-first search and does not use the information provided by the multiplicities.

Also, this modelisation tells us that if h is irreducible, the key reconstruction from the spectrum only has two solutions: h and $\mathrm{h}^\top$.

B Simulation results

ℓ	Syndrome weight		Counters							
			$\in \mathrm{e}^{\perp}$		$\in \mathrm{p}$		$\in \mathrm{c}^{\perp}$		Other	
	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
Average case				$(\in \mathrm{e})$						
-	4 740.826	2 436.014	42.590	17.032	-	-	-	-	28.496	17.045
$\mathcal{S} = \mathcal{N}$										
3	4 750.178	2 447.685	42.629	16.986	41.897	16.997	29.399	17.034	28.550	17.049
4	4 740.988	2 474.073	42.610	17.016	41.982	16.979	29.316	17.056	28.495	17.044
5	4 737.927	2 438.134	42.614	17.027	41.864	17.043	29.404	17.028	28.476	17.041
10	4 724.176	2 441.586	42.698	17.015	40.921	16.975	30.342	17.020	28.390	17.025
30	4 557.218	2 331.555	43.731	16.769	37.008	16.774	34.229	16.775	27.373	16.803
40	4 388.433	2 212.589	44.765	16.516	34.588	16.535	36.677	16.522	26.351	16.553
50	4 138.419	2 032.129	46.288	16.080	31.413	16.114	39.908	16.052	24.841	16.127
60	3 775.992	1 779.213	48.498	15.339	26.942	15.375	44.499	15.360	22.658	15.399
70	3 249.590	1 392.664	51.697	14.018	20.312	14.066	51.333	13.981	19.496	14.108
$\mathcal{S} = 2\mathcal{N}$										
7	4 752.599	2 449.347	42.630	16.968	42.236	16.938	29.190	17.047	28.564	17.052
8	4 741.606	2 448.857	42.624	17.003	42.095	17.014	29.226	17.035	28.497	17.042
9	4 737.578	2 448.165	42.627	17.020	41.986	16.993	29.300	17.029	28.472	17.039
10	4 735.105	2 449.520	42.637	17.017	41.873	17.022	29.389	17.044	28.456	17.036
30	4 654.694	2 405.722	43.133	16.925	40.041	17.168	31.092	17.192	27.960	16.933
50	4 475.758	2 310.459	44.238	16.657	38.243	17.160	32.889	17.178	26.869	16.683
70	4 155.064	2 106.282	46.207	16.120	35.868	16.951	35.277	16.937	24.923	16.154
90	3 611.695	1 745.599	49.530	14.956	32.098	16.281	39.105	16.274	21.640	15.016
110	2 689.331	1 088.797	55.151	12.267	25.314	14.254	46.023	14.168	16.087	12.408
130	1 082.673	214.484	64.897	5.562	12.161	7.660	59.483	7.311	6.458	5.843
$\mathcal{S} = \mathcal{C}$										
5	4 743.816	2 439.823	42.606	17.008	42.282	17.033	28.940	17.071	28.512	17.047
6	4 739.387	2 462.484	42.607	17.010	42.168	17.022	29.013	17.083	28.484	17.044
7	4 737.776	2 453.670	42.614	17.017	42.076	17.017	29.106	17.092	28.474	17.041
8	4 736.235	2 452.988	42.619	17.024	41.981	17.070	29.200	17.098	28.464	17.039
30	4 653.611	2 393.044	43.144	16.916	40.012	17.184	31.132	17.199	27.953	16.933
50	4 471.595	2 304.922	44.265	16.655	38.168	17.163	32.969	17.171	26.843	16.678
70	4 144.716	2 105.496	46.267	16.105	35.726	16.941	35.423	16.945	24.860	16.135
90	3 589.383	1 718.329	49.668	14.901	31.815	16.259	39.391	16.222	21.504	14.963
110	2 642.406	1 058.521	55.436	12.109	24.740	14.157	46.614	14.061	15.804	12.251
130	983.970	184.843	65.497	5.057	10.924	7.260	60.756	6.885	5.865	5.355

Table 3: Influence of codewords and near-codewords on the syndrome weight and the counters distributions for $(r, d, t) = (11\,779, 71, 134)$.

i	ρ_i	i	ρ_i
0	8362.69	18	$6.10582 \cdot 10^{-21}$
1	3255.11	19	$1.01009 \cdot 10^{-22}$
2	619.768	20	$1.56334 \cdot 10^{-24}$
3	76.9735	21	$2.26679 \cdot 10^{-26}$
4	7.02118	22	$3.08321 \cdot 10^{-28}$
5	0.502459	23	$3.9388 \cdot 10^{-30}$
6	0.0294468	24	$4.73139 \cdot 10^{-32}$
7	0.00145736	25	$5.34965 \cdot 10^{-34}$
8	$6.23562 \cdot 10^{-5}$	26	$5.69865 \cdot 10^{-36}$
9	$2.34996 \cdot 10^{-6}$	27	$5.7237 \cdot 10^{-38}$
10	$7.91778 \cdot 10^{-8}$	28	$5.42423 \cdot 10^{-40}$
11	$2.41362 \cdot 10^{-9}$	29	$4.85294 \cdot 10^{-42}$
12	$6.71828 \cdot 10^{-11}$	30	$4.10094 \cdot 10^{-44}$
13	$1.71931 \cdot 10^{-12}$	31	$3.2744 \cdot 10^{-46}$
14	$4.06545 \cdot 10^{-14}$	32	$2.47097 \cdot 10^{-48}$
15	$8.91343 \cdot 10^{-16}$	33	$1.76267 \cdot 10^{-50}$
16	$1.8166 \cdot 10^{-17}$	34	$1.18871 \cdot 10^{-52}$
17	$3.44818 \cdot 10^{-19}$	35	$7.5786 \cdot 10^{-55}$

Table 4: Error factor ρ_i for $(r, d, t) = (12\,323, 71, 134)$.

m	π_m	$p_{\geq m}$	$p_{=m}$
0	0.667	1.0	0.0
1	0.272	1.0	0.0
2	0.0539	1.0	0.0
3	0.00692	1.0	0.0186
4	0.000647	0.981	0.73
5	$4.69 \cdot 10^{-5}$	0.251	0.234
6	$2.75 \cdot 10^{-6}$	0.0168	0.016
7	$1.34 \cdot 10^{-7}$	0.000827	0.000793
8	$5.55 \cdot 10^{-9}$	$3.42 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$
9	$1.98 \cdot 10^{-10}$	$1.22 \cdot 10^{-6}$	$1.18 \cdot 10^{-6}$
10	$6.13 \cdot 10^{-12}$	$3.78 \cdot 10^{-8}$	$3.68 \cdot 10^{-8}$
11	$1.67 \cdot 10^{-13}$	$1.03 \cdot 10^{-9}$	$1.01 \cdot 10^{-9}$
12	$4.05 \cdot 10^{-15}$	$2.49 \cdot 10^{-11}$	$2.44 \cdot 10^{-11}$
13	$8.74 \cdot 10^{-17}$	$5.39 \cdot 10^{-13}$	$5.28 \cdot 10^{-13}$
14	$1.69 \cdot 10^{-18}$	$1.04 \cdot 10^{-14}$	$1.02 \cdot 10^{-14}$

Table 5: Numerical application of Corollary 44 for $(r, d) = (12\,323, 71)$. With Corollary 44 settings, π_m is the probability of having a multiplicity m for a given distance, $p_{\geq m}$ (resp. $p_{=m}$) is the probability of having one distance with multiplicity at least (resp. exactly) m in the whole spectrum.

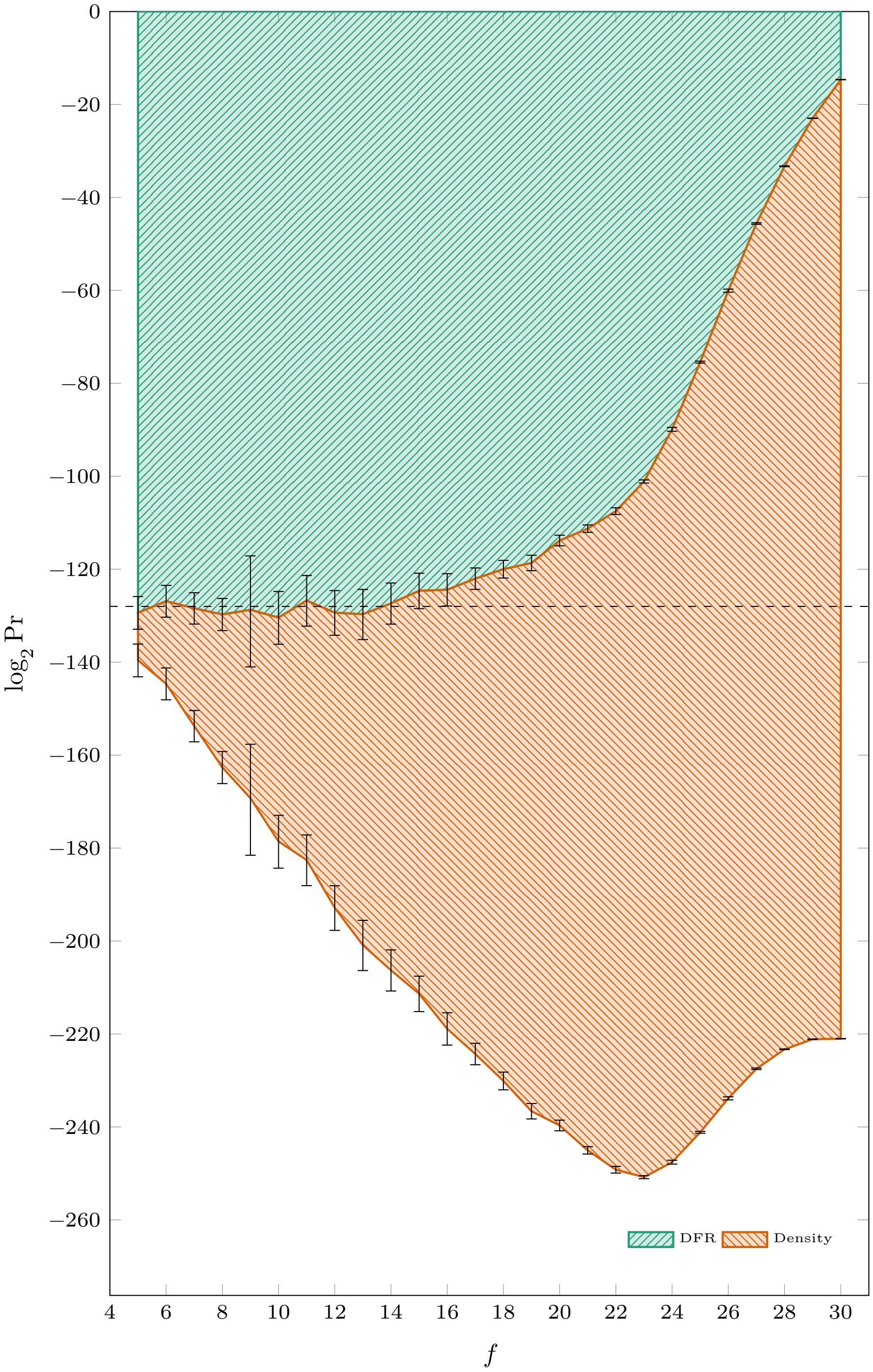

Figure 10: Extrapolated (for $r = 12\,323$) DFR *vs.* number of consecutive ones f with Type I weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (71, 134)$. 99%-confidence intervals.

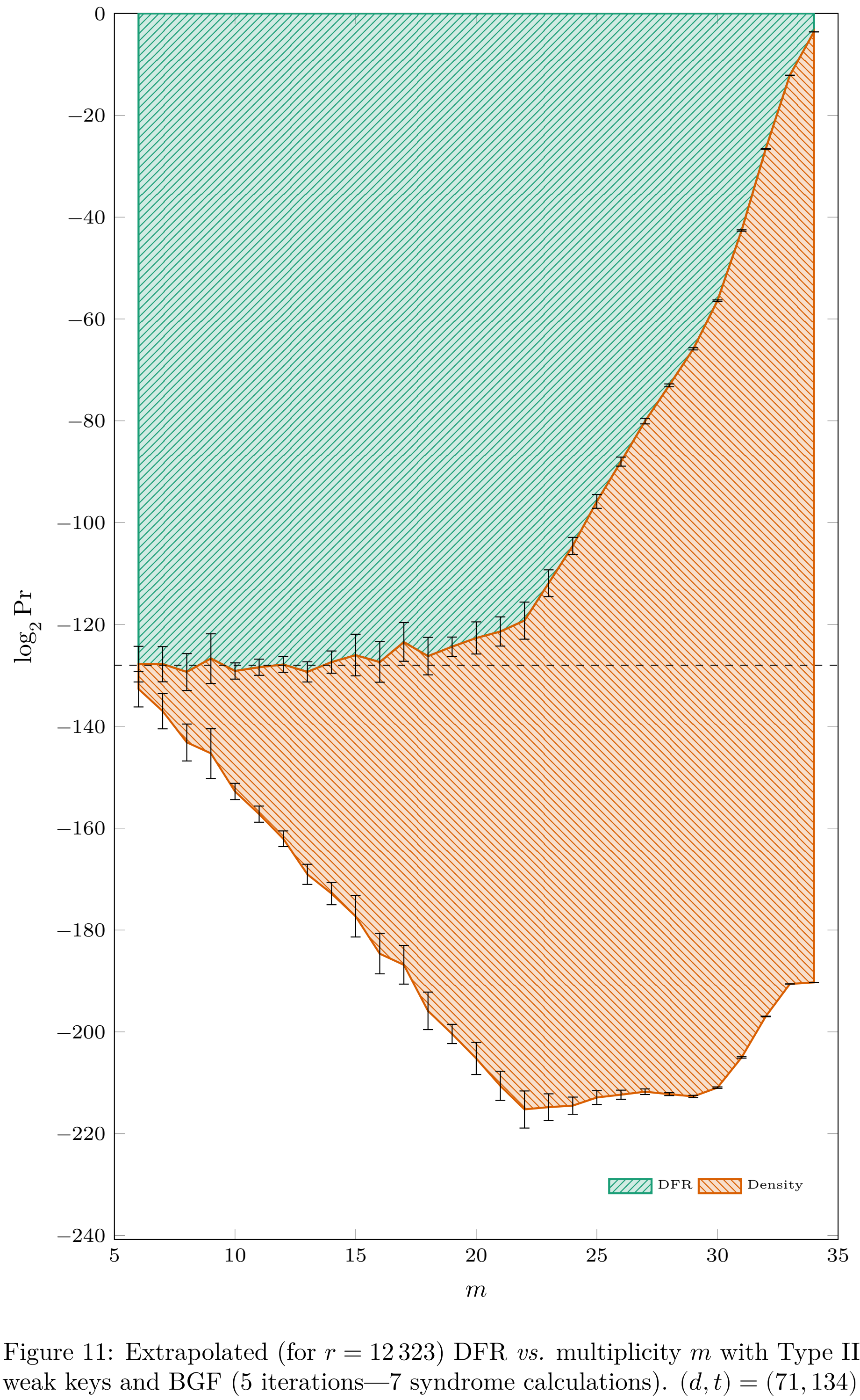

Figure 11: Extrapolated (for $r = 12\,323$) DFR *vs.* multiplicity m with Type II weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (71, 134)$. 99%-confidence intervals.

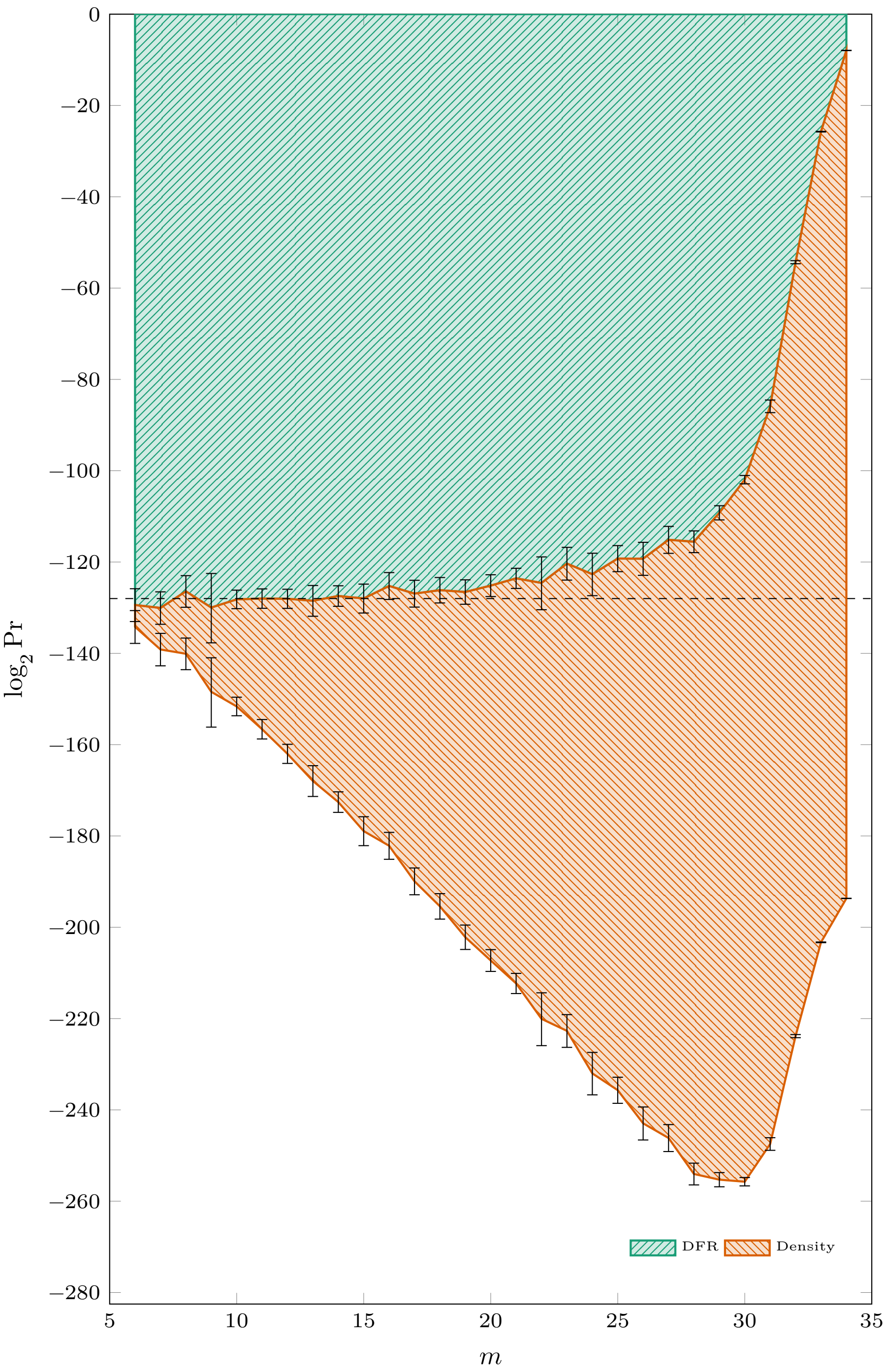

Figure 12: Extrapolated (for $r = 12\,323$) DFR *vs.* multiplicity m with Type III weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (71, 134)$. 99%-confidence intervals.

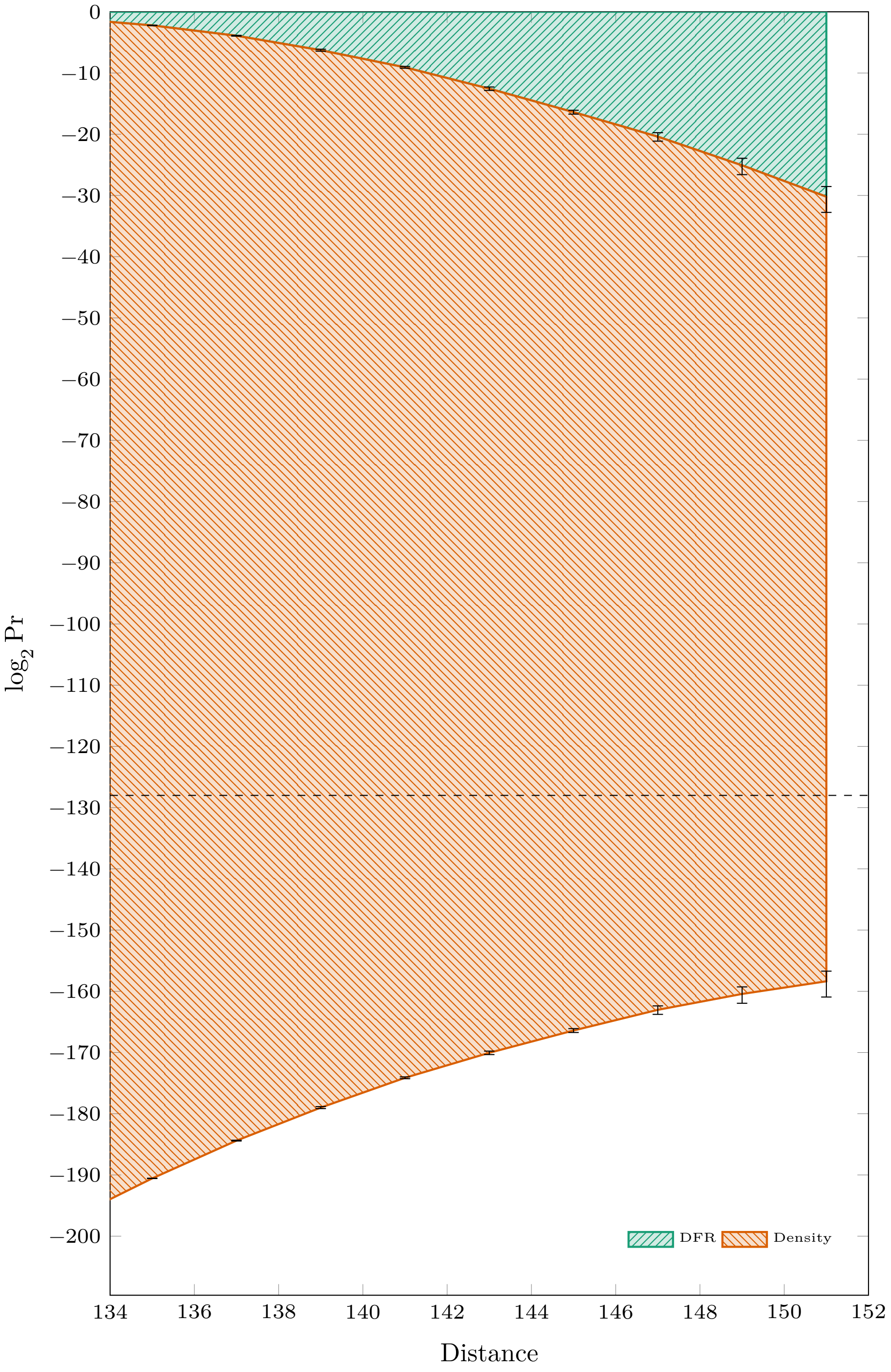

Figure 13: DFR *vs.* distance to $\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (12\,323, 71, 134)$. 99%-confidence intervals.

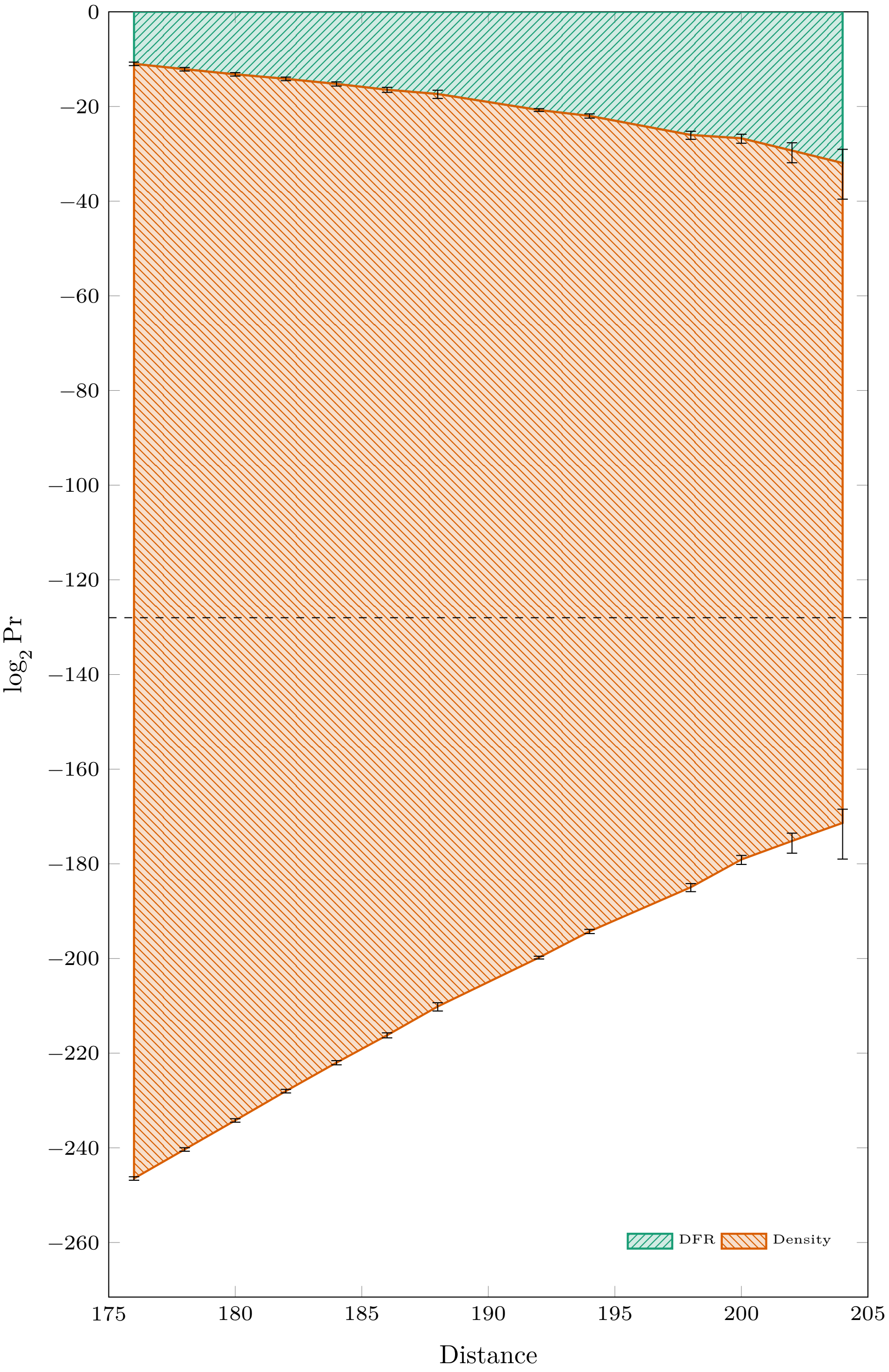

Figure 14: DFR *vs.* distance to $2\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (12\,323, 71, 134)$. 99%-confidence intervals.

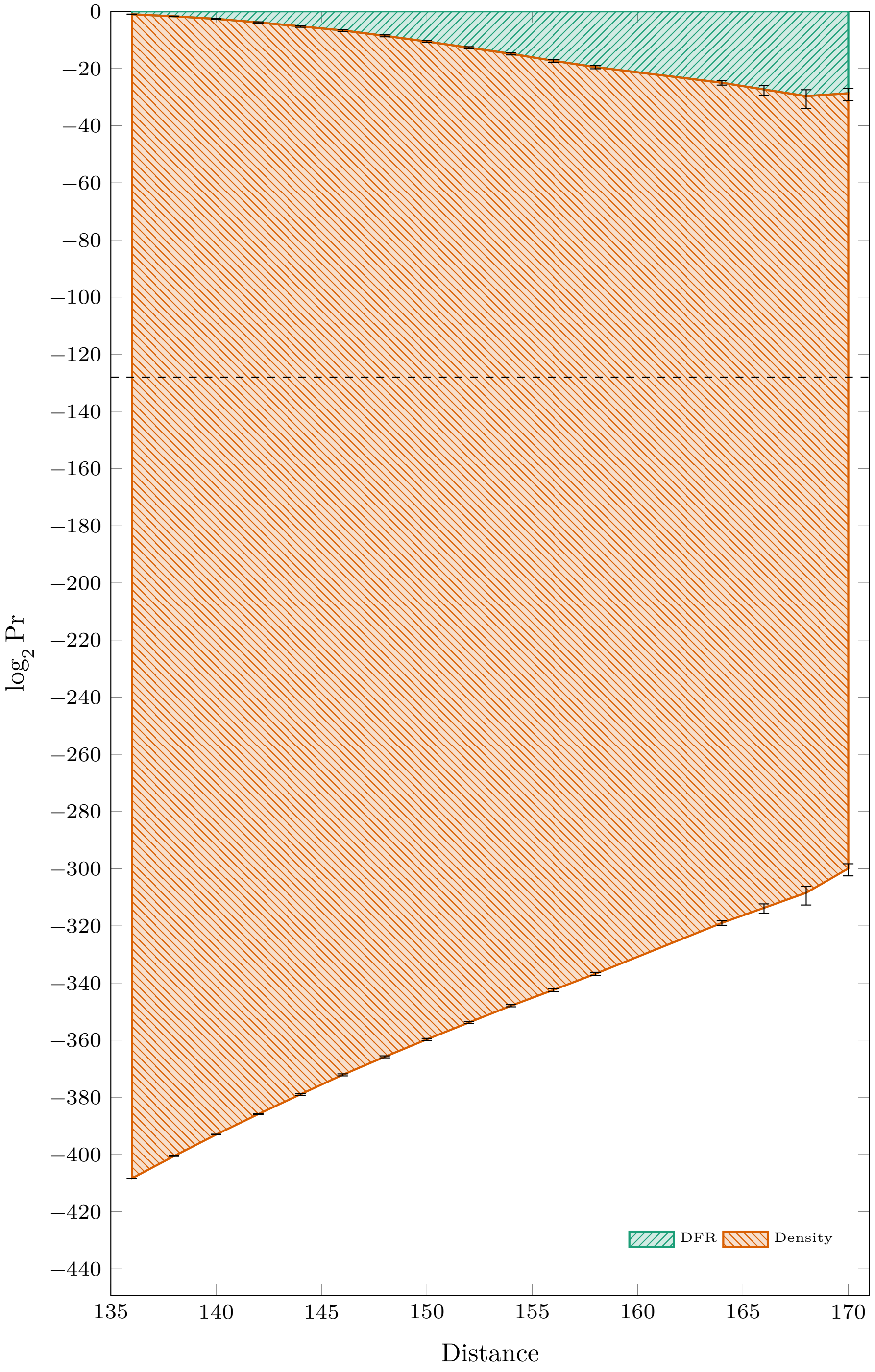

Figure 15: DFR *vs.* distance to $\mathcal{C}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (12\,323, 71, 134)$. 99%-confidence intervals.

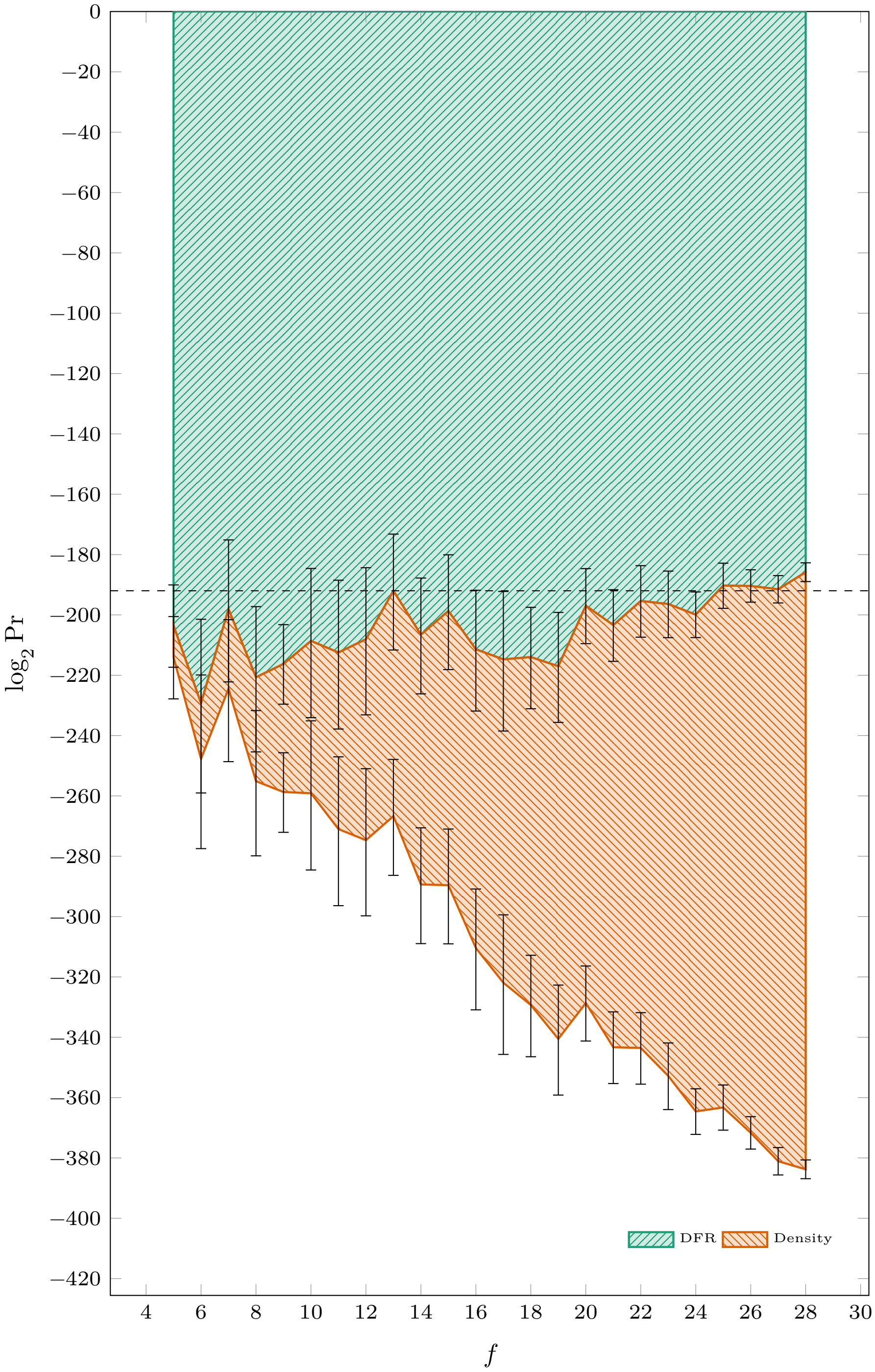

Figure 16: Extrapolated (for $r = 24\,659$) DFR *vs.* number of consecutive ones f with Type I weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (103, 199)$. 99%-confidence intervals.

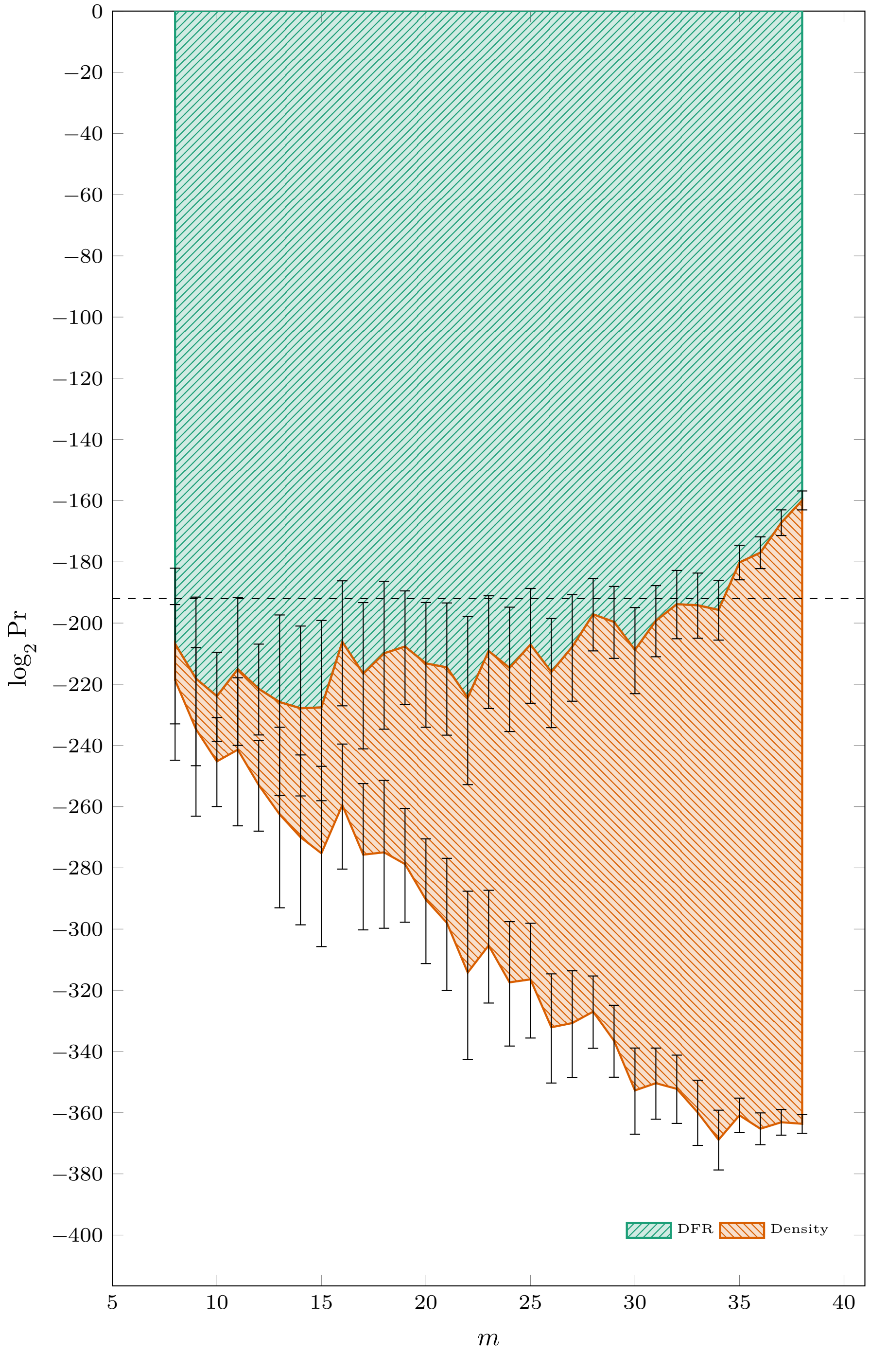

Figure 17: Extrapolated (for $r = 24\,659$) DFR *vs.* multiplicity m with Type II weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (103, 199)$. 99%-confidence intervals.

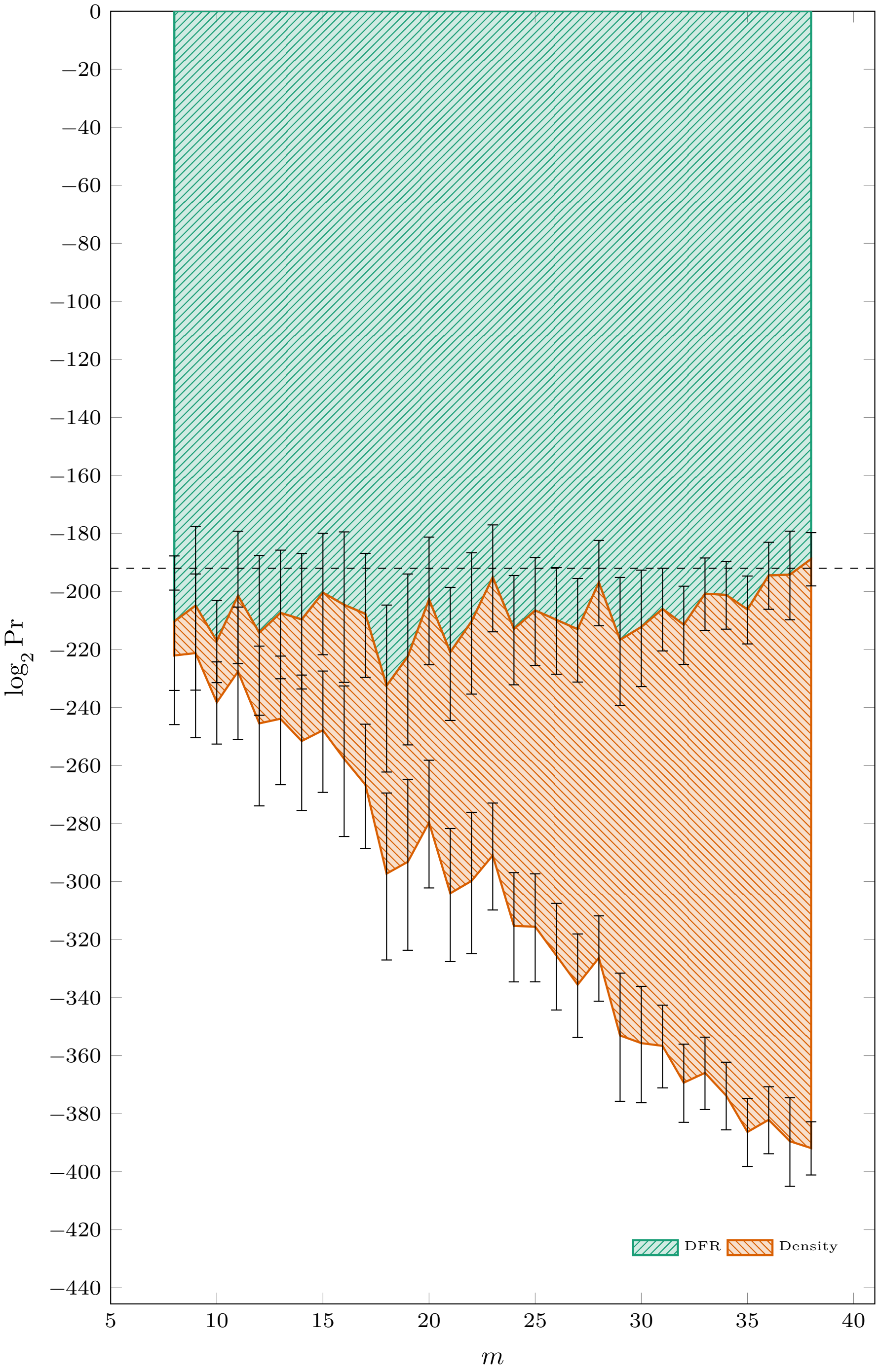

Figure 18: Extrapolated (for $r = 24\,659$) DFR *vs.* multiplicity m with Type III weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (103, 199)$. 99%-confidence intervals.

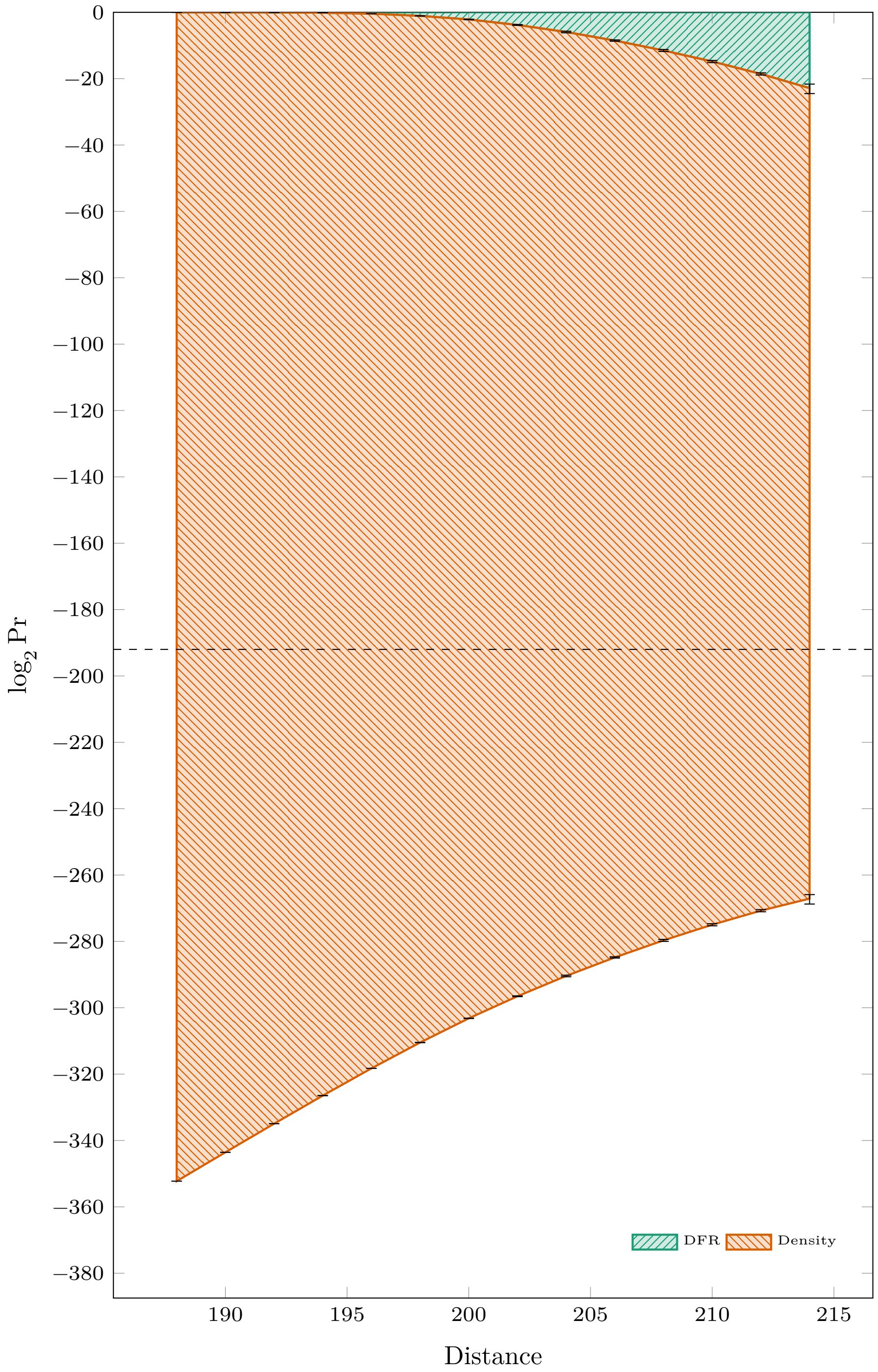

Figure 19: DFR *vs.* distance to $\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (24\,659, 103, 199)$. 99%-confidence intervals.

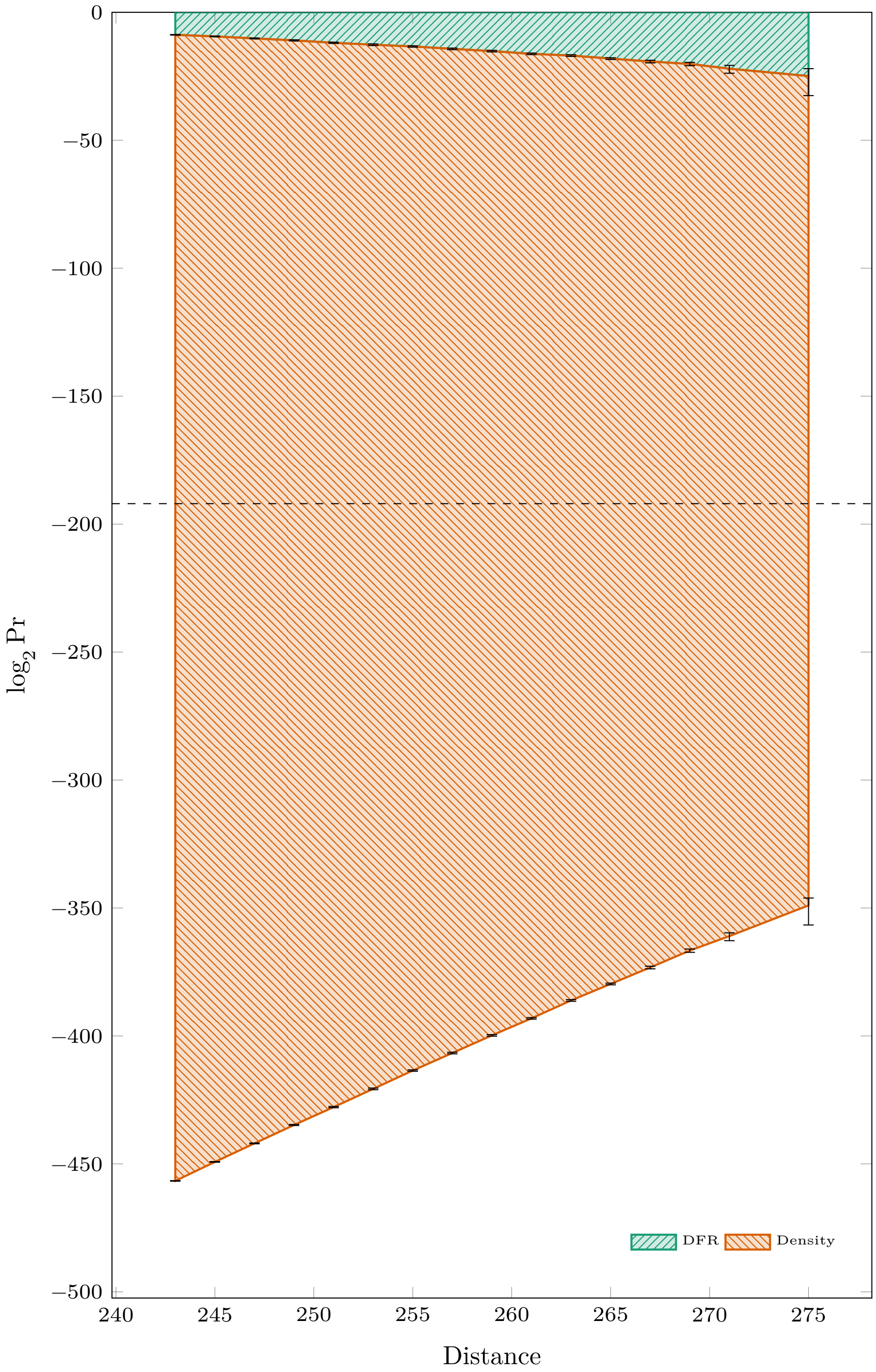

Figure 20: DFR *vs.* distance to $2\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (24\,659, 103, 199)$. 99%-confidence intervals.

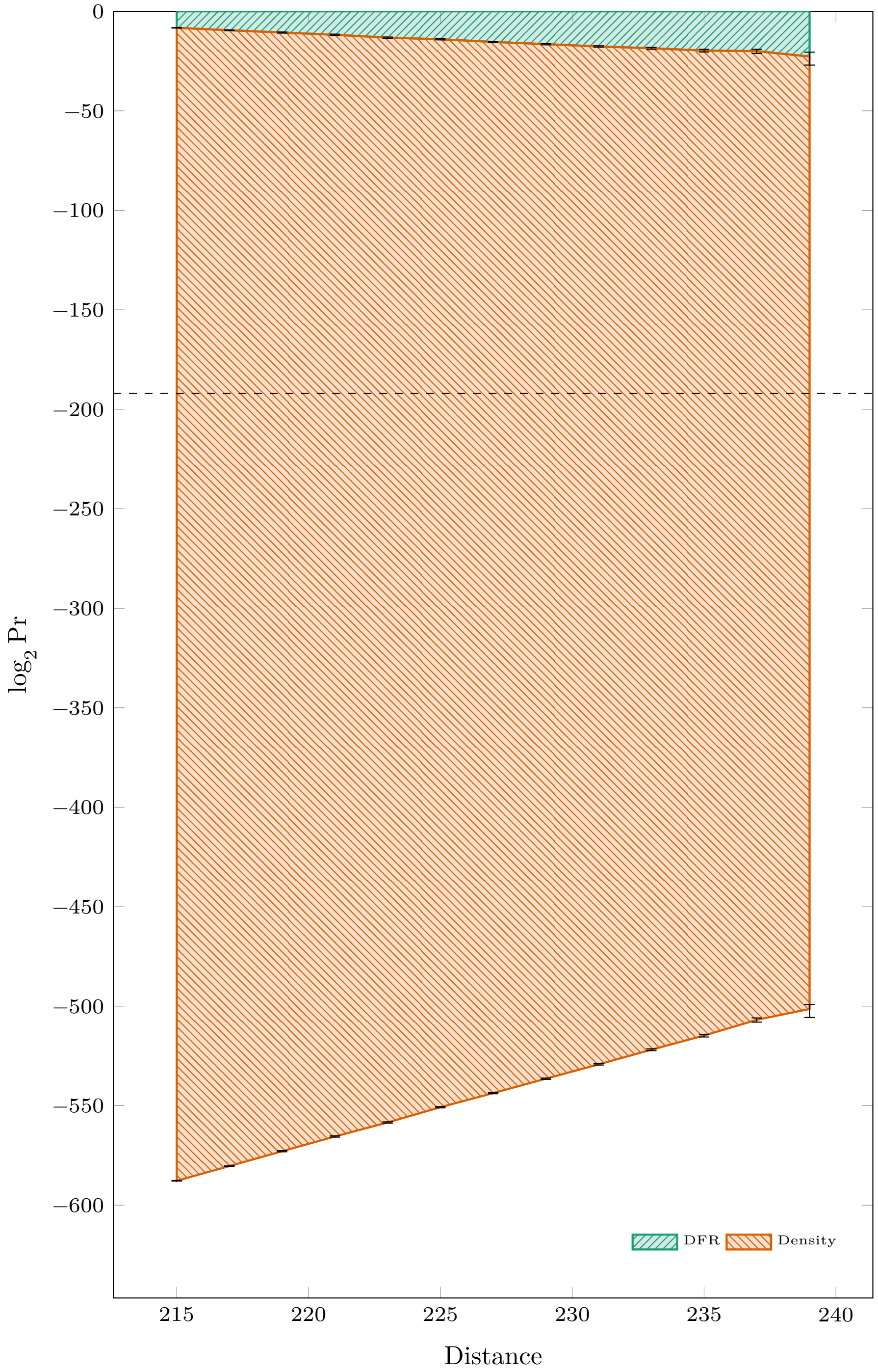

Figure 21: DFR *vs.* distance to $\mathcal{C}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (24\,659, 103, 199)$. 99%-confidence intervals.

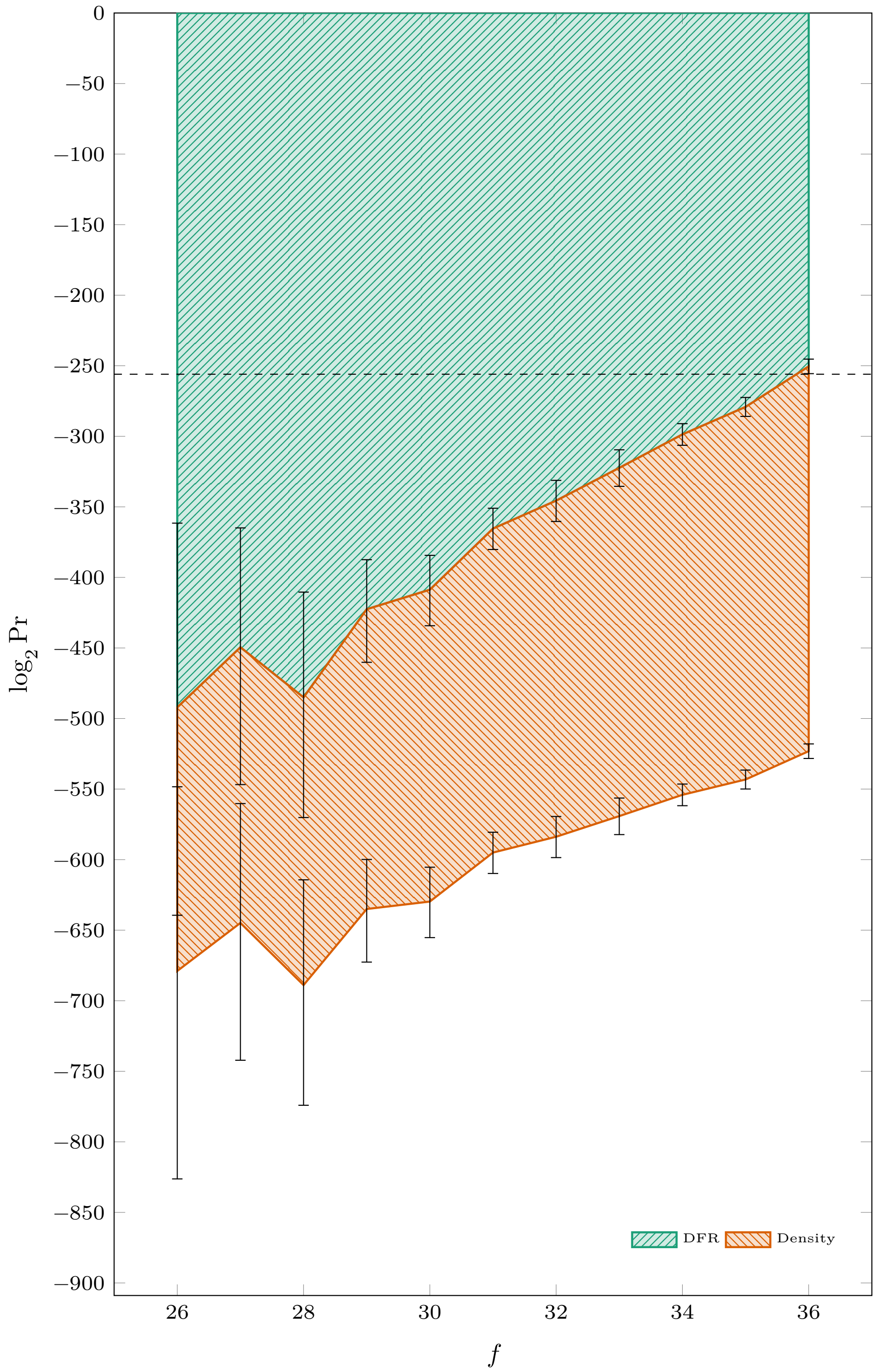

Figure 22: Extrapolated (for $r = 40\,973$) DFR *vs.* number of consecutive ones f with Type I weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (137, 264)$. 99%-confidence intervals.

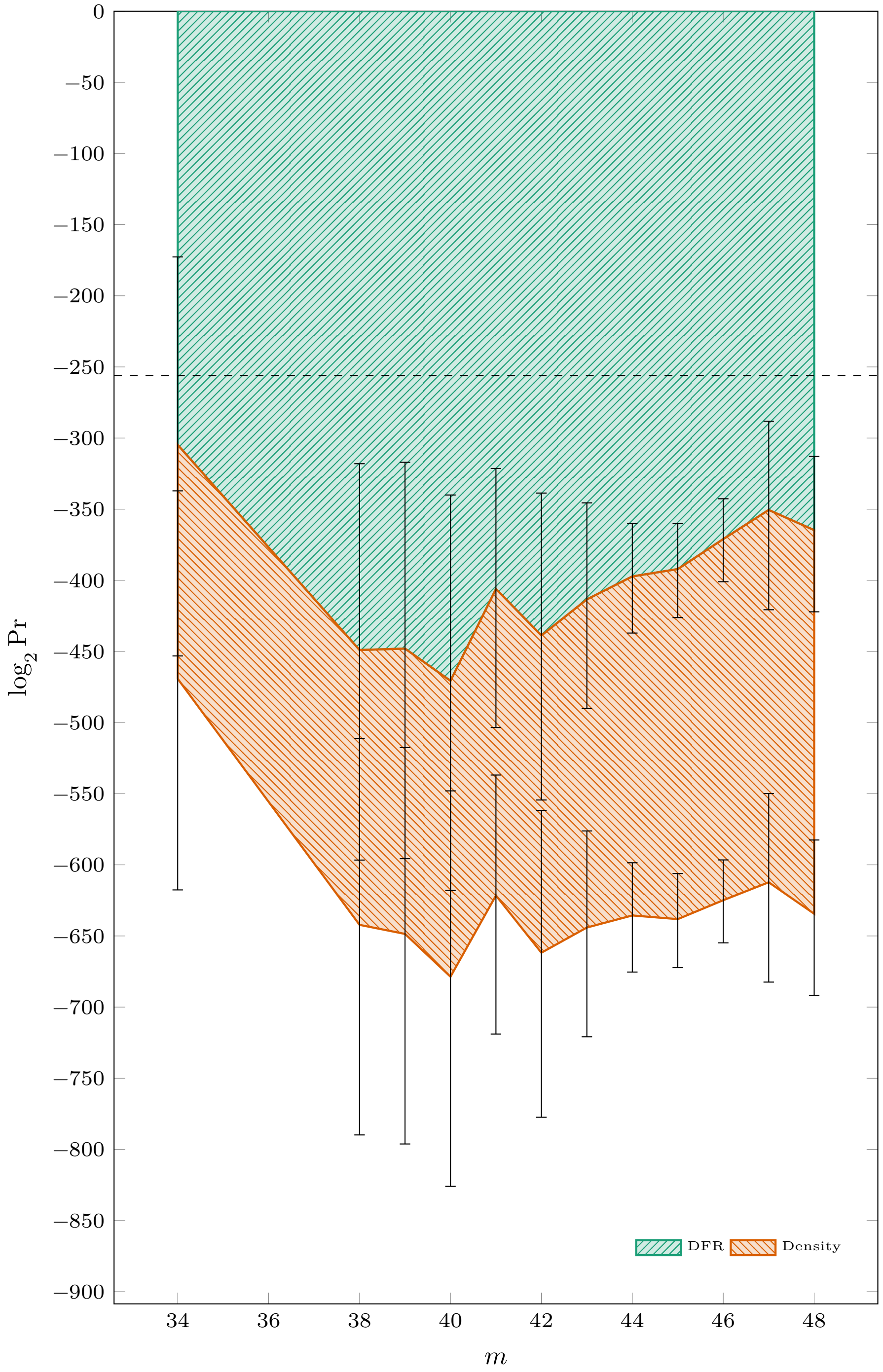

Figure 23: Extrapolated (for $r = 40\,973$) DFR *vs.* multiplicity m with Type II weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (137, 264)$. 99%-confidence intervals.

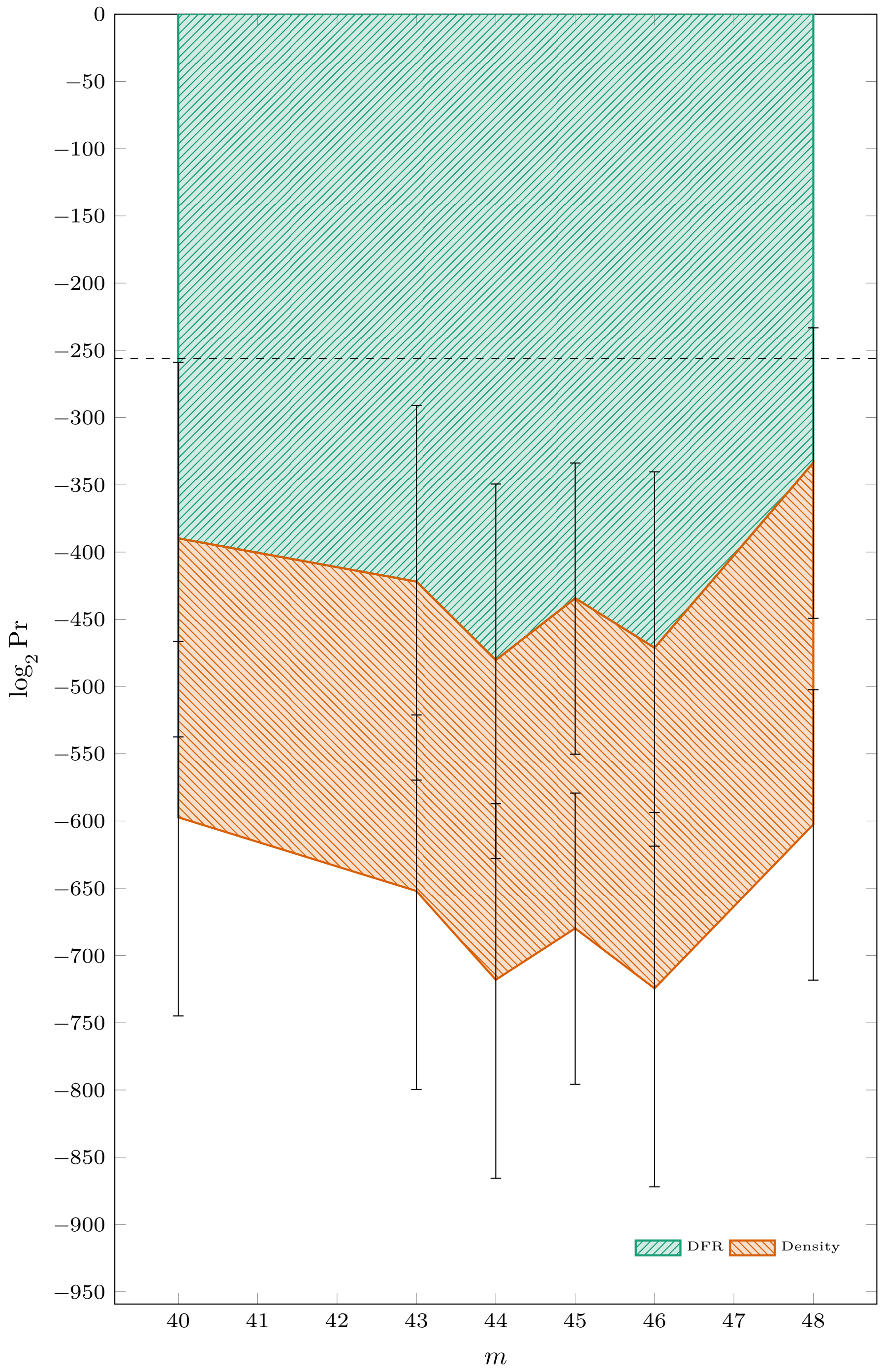

Figure 24: Extrapolated (for $r = 40\,973$) DFR *vs.* multiplicity m with Type III weak keys and BGF (5 iterations—7 syndrome calculations). $(d, t) = (137, 264)$. 99%-confidence intervals.

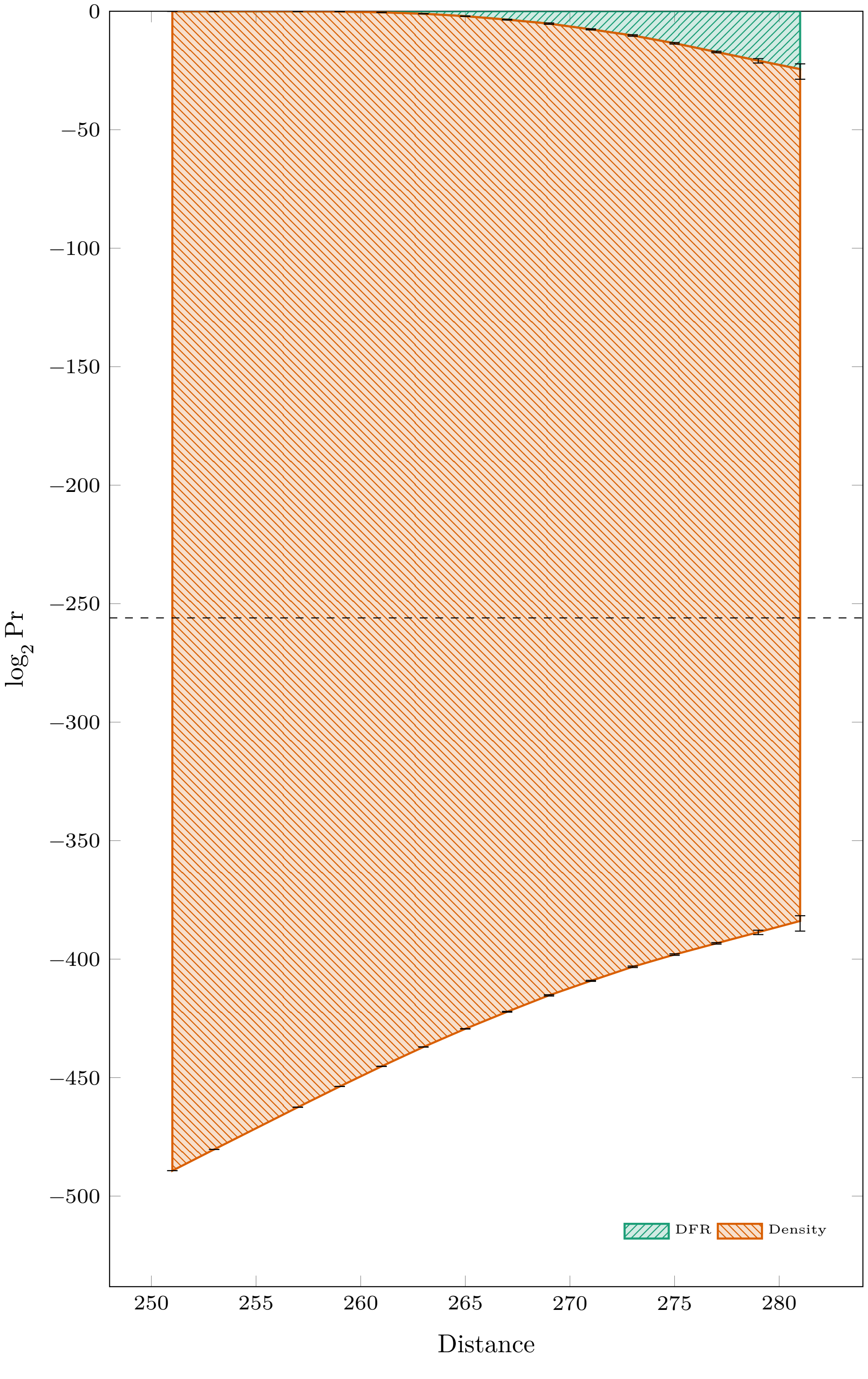

Figure 25: DFR *vs.* distance to $\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (40\,973, 137, 264)$. 99%-confidence intervals.

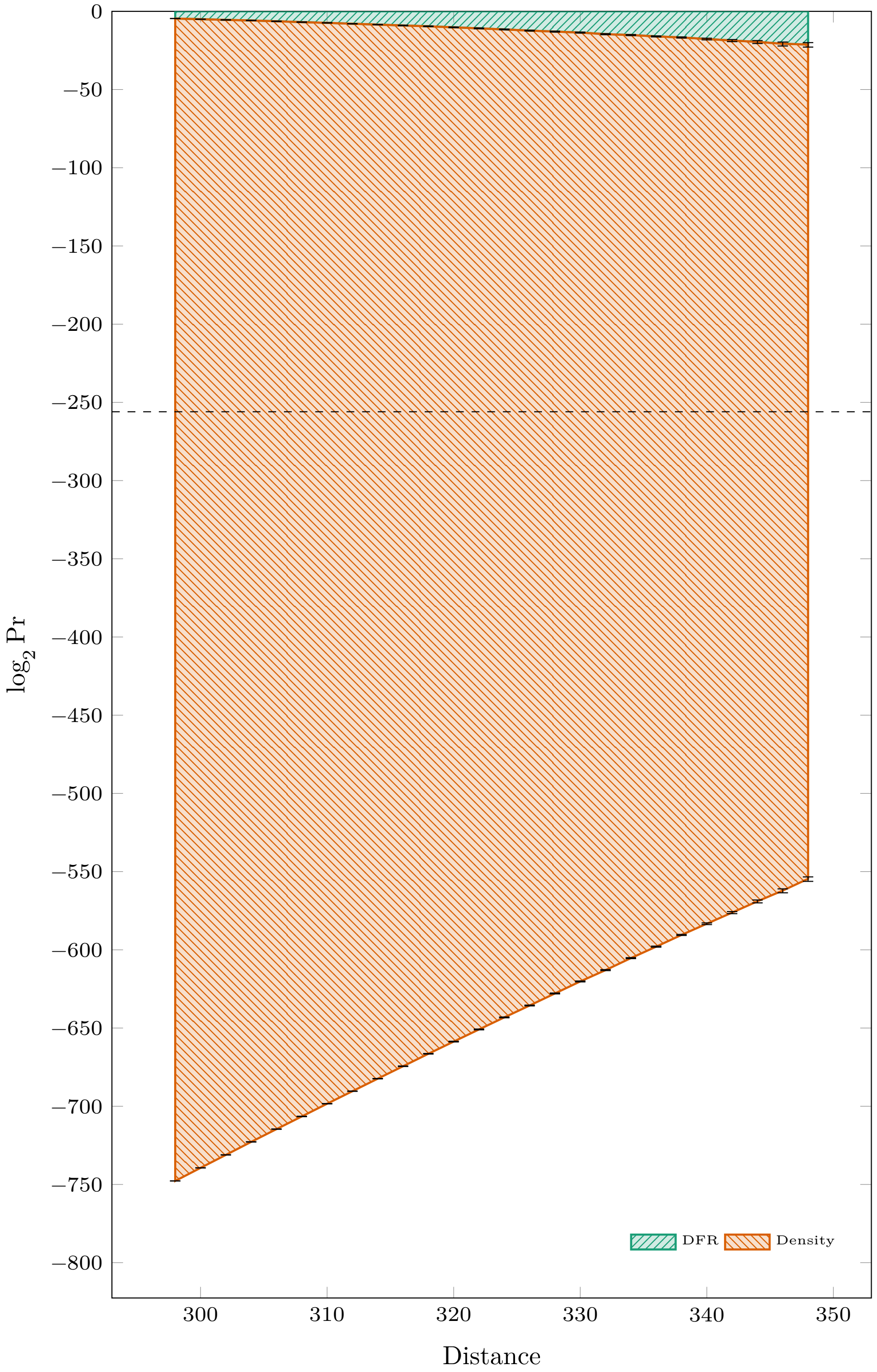

Figure 26: DFR *vs.* distance to $2\mathcal{N}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (40\,973, 137, 264)$. 99%-confidence intervals.

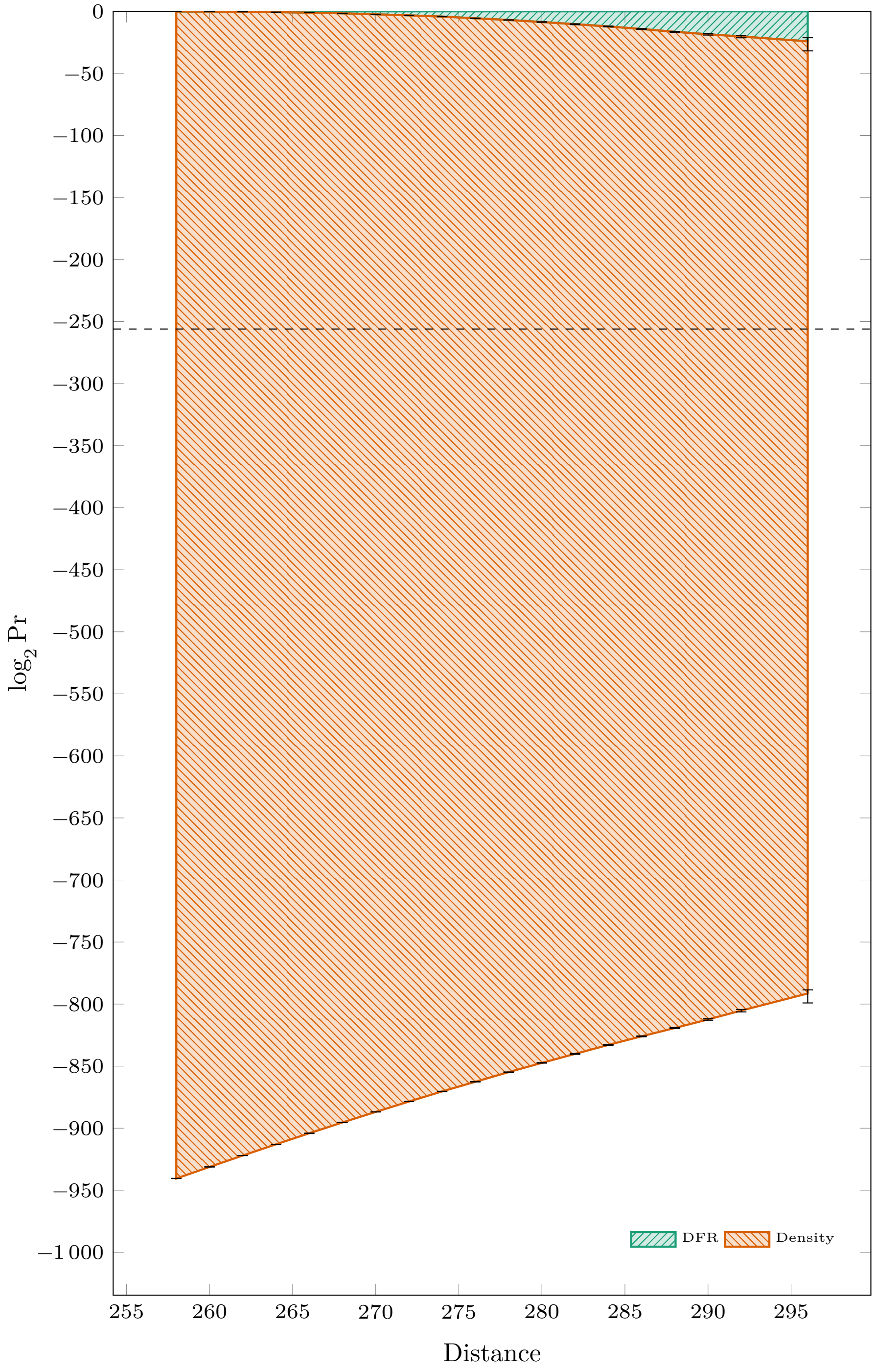

Figure 27: DFR *vs.* distance to $\mathcal{C}$ with BGF (5 iterations—7 syndrome calculations). $(r, d, t) = (40\,973, 137, 264)$. 99%-confidence intervals.