Top-m identification for linear bandits

Clémence Réda[1,*] **Emilie Kaufmann**[2] **Andrée Delahaye-Duriez**[1,3,4]

[1] Université de Paris, Inserm UMR 1141 NeuroDiderot, F-75019, Paris, France, *clemence.reda@inria.fr
[2] Université Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[3] Université Sorbonne Paris Nord, UFR SMBH, F-93000, Bobigny, France & [4] Assistance Publique des Hôpitaux de Paris, Hôpital Jean Verdier, Service d'Histologie-Embryologie-Cytogénétique, F-93140, Bondy, France

Abstract

Motivated by an application to drug repurposing, we propose the first algorithms to tackle the identification of the $m \geq 1$ arms with largest means in a linear bandit model, in the fixed-confidence setting. These algorithms belong to the generic family of Gap-Index Focused Algorithms (GIFA) that we introduce for Top-m identification in linear bandits. We propose a unified analysis of these algorithms, which shows how the use of features might decrease the sample complexity. We further validate these algorithms empirically on simulated data and on a simple drug repurposing task.

1 INTRODUCTION

The multi-armed bandit setting, in which an agent sequentially gathers samples from K unknown probability distributions called arms, is a powerful framework for sequential resource allocation tasks. While a large part of the literature focuses on the reinforcement learning problem in which the samples are viewed as reward that the agent seeks to maximize (Bubeck and Cesa-Bianchi, 2012), *pure-exploration* objectives have also received a lot of attention (Bubeck et al., 2009; Degenne and Koolen, 2019). In this paper, we focus on *Top-m identification* in which the goal is to identify the $m < K$ arms with the largest expected rewards. While several Top-m identification algorithms have been given, no algorithm has been specifically designed to tackle the challenging *linear bandit* setting (Auer, 2003). This paper aims at filling this gap.

In a linear bandit model, the mean μ_a of each arm a is assumed to depend linearly on a known feature vector $x_a \in \mathbb{R}^N$ associated to the arm: $\mu_a = \theta^\top x_a$ for some vector $\theta \in \mathbb{R}^N$. In contrast, the so-called classical bandit model does not make any assumption on the means $(\mu_a)_{a \in [K]}$[1]. A Top-m identification algorithm outputs a subset of size m as a guess for the m arms with largest means. Several objectives exist: in the fixed-budget setting, the goal is to minimize the probability that the guess is wrong after a pre-specified total number of samples from the arms (Bubeck et al., 2013). In this paper, we focus on the *fixed-confidence setting*, in which the error probability should be guaranteed to be smaller than a given risk parameter $\delta \in (0, 1)$ while minimizing the *sample complexity*, that is, the total number of samples needed to output the guess. This choice is motivated by a real-life application to drug repurposing, in which we would like to control the failure rate in our predictions.

Drug repurposing is a field of research aimed at discovering new indications for drugs which are already approved for marketing, and may contribute to solve the problem of ever increasing research budget need for drug discovery (Hwang et al., 2016). For a given disease, we are interested in identifying a subset of drugs that may have a therapeutic interest. Providing a group of 5 or 10 drugs, rather than a single one, can ease the decision of further investigation, as many leads are provided. We believe sequential methods could be of interest, when considering a drug repurposing method called "signature reversion" (Musa et al., 2018). In this context, drugs recommended for repurposing are the ones minimizing the difference in gene activity between treated patients and healthy individuals. A possible way to measure this difference is to build a simulator that evaluates the genewise impact of a given drug on patients. However, this simulator may be stochastic and computationally expen-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

[1] For $n \in \mathbb{N}^*$ we use the shorthand $[n] = \{1, \dots, n\}$.

sive hence the need for sequential queries, that can be modeled as sampling of arms. As the arms (drugs) can be characterized by a real-valued feature vector which represents genewise activity change due to treatment, we resort to linear bandits to tackle this Top-m identification problem.

Related work Two types of fixed-confidence algorithms have been proposed for Top-m identification in a classical bandit: those based on adaptive sampling such as LUCB (Kalyanakrishnan et al., 2012) or UGapE (Gabillon et al., 2012) or those based on uniform sampling and eliminations (Kaufmann and Kalyanakrishnan, 2013; Chen et al., 2017). For linear bandits, to the best of our knowledge, the only efficient algorithms have been proposed for the *best arm identification* (BAI) problem, which corresponds to $m = 1$. This setting, first investigated by Soare et al. (2014), recently received a lot of attention: an efficient adaptive sampling algorithm called LinGapE was proposed by Xu et al. (2018) and subsequent work such as Fiez et al. (2019) sought to achieve the minimal sample complexity. In particular, the LinGame algorithm of Degenne et al. (2020) is proved to exactly achieve the problem-dependent sample complexity lower bound for linear BAI in a regime in which δ goes to zero.

We note that, in principle, LinGame can be used for any pure exploration problem in a linear bandit, which includes Top-m identification for $m > 1$. However, this algorithm uses a game theoretic formalism which needs the computation of a best response for Nature in response to the player's selection; a computable expression of this strategy is not available to our knowledge for Top-m ($m > 1$). Besides, computing the information-theoretic lower bound for Top-m identification is also computationally hard. These remarks led us to investigate efficient adaptive sampling algorithms for general Top-m identification ($m \geq 1$) in linear bandit, which are still missing in the literature, instead of trying to propose asymptotically optimal algorithms, as done in linear BAI.

Contributions First, by carefully looking at known adaptive sampling bandits for classical Top-m, we propose a generic algorithm structure based on *Gap Indices*, called GIFA, which encompasses existing adaptive algorithms for classical Top-m identification and linear BAI. This structure allows a higher order and modular understanding of the learning process, and correctness properties can readily be inferred from a partially specified bandit algorithm. It allows us to define two interesting new algorithms, called m-LinGapE and LinGIFA. In Section 4, we present a unified sample complexity analysis of a subclass of GIFA algorithms which comprises existing methods, which shows that the use of features can help decreasing the sample complexity in some cases. Finally, we show in Section 5 that m-LinGapE and LinGIFA perform better than their counterparts for classical bandits, both on artificially generated linear bandit instances and on a simple instance of our drug repurposing application, that is described in detail in Appendix B.

Notation We let $\overset{[m]}{\max}$, $\overset{m}{\max}$ (resp. $\overset{[m]}{\min}$, $\overset{m}{\min}$) be the operator returning the m, m^{th} greatest (resp. smallest) value(s), $\|x\|_M = \sqrt{x^\top M x}$ where M is positive definite, and $\|x\| = \sqrt{x^\top x}$.

2 SETTING

In this section, we introduce the *probably approximately correct* (PAC) fixed-confidence linear Top-m problem. From now on, we identify each arm by an integer in $[K]$ such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m > \mu_{m+1} \geq \mu_{m+2} \geq \cdots \geq \mu_K$. This ordering is of course unknown to the learner. In the PAC formalism, we are willing to relax the optimality constraint on arms using slack variable $\varepsilon > 0$ and we denote by $\mathcal{S}_m^{*,\varepsilon} \triangleq \{a \in [K] : \mu_a \geq \mu_m - \varepsilon\}$ the set of best arms up to ε. Since $\mu_m > \mu_{m+1}$, we use the shorthand $\mathcal{S}_m^\star \triangleq \mathcal{S}_m^{*,0} = [m]$ for the set of m arms with largest means. The set of $K - m$ arms worst arms is its complementary $(\mathcal{S}_m^\star)^c = [K] \backslash \mathcal{S}_m^\star$. A linear bandit model is parameterized by an unknown vector $\theta \in \mathbb{R}^N$, such that there are constants $L, S > 0$ which satisfy $\|\theta\| \leq S \in \mathbb{R}^{*+}$ and such that the mean of arm a is $\mu_a = \theta^\top x_a$, where the feature vector x_a satisfies $\|x_a\| \leq L \in \mathbb{R}^{*+}$. In each round $t \geq 1$, a learner selects an arm $a_t \in [K]$ and observes a sample

$$r_t = \theta^\top x_{a_t} + \eta_t,$$

where η_t is a zero-mean sub-Gaussian noise with variance σ^2 –that is, $\mathbb{E}[e^{\lambda \eta_t}] \leq \exp\left((\lambda^2\sigma^2)/2\right)$ for all $\lambda \in \mathbb{R}$ – which is independent from past observations.

For $m \in [K]$, an algorithm for Top-m identification consists of a *sampling rule* $(a_t)_{t \in \mathbb{N}^*}$, where a_t is $\mathcal{F}_{t-1}$-measurable, and $\mathcal{F}_t = \sigma(a_1, r_1, \ldots, a_t, r_t)$ is the σ-algebra generated by the observations up to round t ; a *stopping rule* τ, which is a stopping time with respect to $\mathcal{F}_t$ that indicates when the learning process is over; and a *recommendation rule* $\hat{S}_m^\tau$, which is a $\mathcal{F}_\tau$-measurable set of size m that provides a guess for the m arms with the largest means. Then, given a slack variable $\varepsilon \geq 0$ and the failure rate $\delta \in (0, 1)$, an (ε, m, δ)-PAC algorithm is such that

$$\mathbb{P}\left(\hat{S}_m^\tau \subseteq \mathcal{S}_m^{*,\varepsilon}\right) \geq 1 - \delta.$$

In the fixed confidence setting, the goal is to design a (ε, m, δ)-PAC algorithm with a small *sample complexity* τ. In order to do so, the learner needs to estimate the unknown parameter $\theta \in \mathbb{R}^N$, which can be done with a (regularized) least-squares estimator. For any arm $a \in [K]$, we let $N_a(t) \triangleq \sum_{s=1}^{t} \mathbb{1}(a_s = a)$ be the number of times that arm a is sampled up to time t, and define the λ-regularized design matrix and least-squares estimate:

$$\hat{V}_t^\lambda \;\triangleq\; \lambda I_N + \sum_{a=1}^{K} N_a(t) x_a x_a^\top$$

$$\text{and } \hat{\theta}_t^\lambda \;\triangleq\; \left(\hat{V}_t^\lambda\right)^{-1} \left(\sum_{s=1}^{t} r_s x_{a_s}\right) .$$

We let $\hat{\Sigma}_t^\lambda \triangleq \sigma^2 \left(\hat{V}_t^\lambda\right)^{-1}$, which can be interpreted as the posterior covariance in a Bayesian linear regression model in which the covariance of the prior is $(\sigma^2/\lambda) I_N$.

The algorithms that we present in the next section crucially rely on estimating the gaps $\Delta_{i,j} \triangleq \mu_i - \mu_j$ between pairs of arms (i,j), and building *upper confidence bounds* (UCBs) on these quantities. For this purpose, we introduce the empirical mean of each arm a, $\hat{\mu}_a(t) \triangleq (\hat{\theta}_t^\lambda)^\top x_a$ and define the empirical gap $\hat{\Delta}_{i,j}(t) \triangleq \hat{\mu}_i(t) - \hat{\mu}_j(t)$. A first option to build UCBs on $\Delta_{i,j}$ consists in building individual confidence intervals on the mean of each arm a, that are of the form $L_a(t) = \hat{\mu}_a(t) - W_t(a)$ for the lower-confidence bound, and $U_a(t) = \hat{\mu}_a(t) + W_t(a)$ for the upper confidence bound, where $W_t(a) \triangleq C_{t,\delta} \|x_a\|_{\hat{\Sigma}_t^\lambda}$ for some threshold function $C_{t,\delta}$ to be specified later. Clearly,

$$B_{i,j}^{\text{ind}}(t) = U_i(t) - L_j(t) = \hat{\Delta}_{i,j}(t) + W_t(i) + W_t(j)$$

is an upper bound on $\Delta_{i,j}$ if $L_j(t) \le \mu_j$ and $\mu_i \le U_i(t)$. Yet, using the linear model, one can also directly build an UCB on the difference via

$$B_{i,j}^{\text{pair}}(t) = \hat{\Delta}_{i,j}(t) + C_{t,\delta} \|x_i - x_j\|_{\hat{\Sigma}_t^\lambda} .$$

Both constructions lead to symmetrical bounds[2] of the form $B_{i,j}(t) = \hat{\Delta}_{i,j}(t) + W_t(i,j)$, where $W_t(i,j) = C_{t,\delta} \|x_i - x_j\|_{\hat{\Sigma}_t^\lambda}$ for paired UCBs and $W_t(i,j) = C_{t,\delta} \left(\|x_i\|_{\hat{\Sigma}_t^\lambda} + \|x_j\|_{\hat{\Sigma}_t^\lambda}\right)$ for individual UCBs. In both cases, $W_t(i,j) = W_t(j,i) \ge 0$. The fact that these quantities are indeed upper confidence bounds (for some pairs $(i,j) \in [K^2]$) will be justified in Section 3.

Remark 1. *Observe that the linear bandit setting subsumes the classical bandit model, choosing arm features $(x_a)_{a \in [K]}$ to be vectors of the canonical basis of $\mathbb{R}^K$ $(N = K)$ and $\theta = \mu = [\mu_1, \mu_2, \ldots, \mu_K]^\top$. Then, $\lambda = 0$ yields standard UCBs in this model: $\hat{\mu}_a(t)$ reduces to the empirical average of the rewards gathered from arm a and $\|x_a\|_{\hat{\Sigma}_t^\lambda} = \sigma/\sqrt{N_a(t)}$.*

3 GAP-INDEX FOCUSED ALGORITHMS (GIFA)

3.1 Generic GIFA Algorithms

Looking at the literature for Top-m identification and BAI in (linear) bandits, existing adaptive sampling designs (Kalyanakrishnan et al., 2012; Gabillon et al., 2012; Xu et al., 2018) have many ingredients in common. We formalize in this section a generic structure that encompasses these algorithms, and under which we propose two new algorithms. We introduce the notion of "Gap-Index Focused" Algorithms (GIFA), which associates to each *pair of arm* (i,j) an index $B_{i,j}(t)$ at time t (called "gap index").

The idea in GIFA is to estimate in each round t a set of candidate m best arms, denoted by $J(t)$, and to select the two most ambiguous arms: $b_t \in J(t)$, which can be viewed as a guess for the m-best arm, and a *challenger* $c_t \notin J(t)$. c_t is defined as a potentially misassessed m-best arm, with largest possible gap to b_t: $c_t = \arg\max_{c \in [K]} B_{c,b_t}(t)$. The idea of using two ambiguous arms goes back to LUCB (Kalyanakrishnan et al., 2012) for Top-m identification in classical bandits. Then the final arm a_t selected by the algorithm should help discriminate between b_t and c_t. A naive idea is to either draw b_t or c_t, but alternative selection rules will be discussed later. At the end of the learning phase, at stopping time τ, the final set $J(\tau)$ is recommended.

Algorithm 1 GIFA for (ε, δ)-PAC Top-m

```
Input: $K$ arms of means $\mu_1, \ldots, \mu_K$; $\varepsilon \ge 0$;
$m \le K; \delta \in (0,1)$.
Output: $\hat{S}_m^{\tau_\delta}$ estimated $m$ $\varepsilon$-optimal arms at final time $\tau_\delta$
$t \leftarrow 1$
initialization()
while $\neg$ stopping_rule($(B_{i,j}(t))_{i,j \in [K]}; \varepsilon, t$) do
    // $J(t)$: estimated $m$ best arms at $t$
    $J(t) \leftarrow$ compute_Jt($(B_{i,j}(t))_{i,j \in [K]}, (\hat{\mu}_i(t))_{i \in [K]}$)
    // $b_t$: estimated $m$-best arm at $t$
    $b_t \leftarrow$ compute_bt($J(t), (B_{i,j}(t))_{i,j \in [K]}$)
    // $c_t$: challenger to $b_t$ at $t$
    $c_t \leftarrow \arg\max_{a \notin J(t)} B_{a,b_t}(t)$
    // selecting and pulling arms
    $a_t \leftarrow$ selection_rule($b_t, c_t; \hat{V}_{t-1}^\lambda$)
    $r_t \leftarrow$ pulling($a_t$)
    Update design matrix $\hat{V}_t^\lambda$, means $(\hat{\mu}_i(t))_{i \in [K]}$
    Update gap indices $(B_{i,j}(t+1))_{i,j \in [K]}$
    $t \leftarrow t + 1$
end while
$\hat{S}_m^{\tau_\delta} \leftarrow J(\tau_\delta)$
return $\hat{S}_m^{\tau_\delta}$
```

[2] Note also that there exist algorithms with non symmetrical bounds, such the KL version of LUCB (Kaufmann and Kalyanakrishnan, 2013).

Table 1: Adaptive samplings for Top-m (our proposals are in bold type ; except for LUCB and UGapE, all algorithms use indices $B = B^{\text{pair}}$ instead of individual indices B^{ind}).

Algorithm	`compute_Jt`	`compute_bt`	`selection_rule`	`stopping_rule`
LUCB	$\underset{j\in[K]}{\overset{[m]}{\arg\max}}\, \hat{\mu}_j(t)$	$\underset{j\in J(t)}{\arg\max}\ \underset{i\notin J(t)}{\max}\ B_{i,j}(t)$	largest variance	$B_{c_t,b_t}(t) \le \varepsilon$
UGapE	$\underset{j\in[K]}{\overset{[m]}{\arg\min}}\, \underset{i\neq j}{\overset{m}{\max}}\, B_{i,j}(t)$	$\underset{j\in J(t)}{\arg\max}\ \underset{i\notin J(t)}{\max}\ B_{i,j}(t)$	largest variance	$\underset{j\in J(t)}{\max}\ \underset{i\neq j}{\overset{m}{\max}}\, B_{i,j}(t) \ \le \varepsilon$
LinGapE	$\underset{j\in[K]}{\arg\max}\, \hat{\mu}_j(t)$	$J(t) \quad (\lvert J(t)\rvert = m = 1)$	greedy, optimized	$B_{c_t,b_t}(t) \le \varepsilon$
***m*-LinGapE** $(m \geq 1)$	$\underset{j\in[K]}{\overset{[m]}{\arg\max}}\, \hat{\mu}_j(t)$	$\underset{j\in J(t)}{\arg\max}\ \underset{i\notin J(t)}{\max}\ B_{i,j}(t)$	largest variance, greedy, optimized	$B_{c_t,b_t}(t) \le \varepsilon$
LinGIFA	$\underset{j\in[K]}{\overset{[m]}{\arg\min}}\, \underset{i\neq j}{\overset{m}{\max}}\, B_{i,j}(t)$	$\underset{j\in J(t)}{\arg\max}\, \underset{i\neq j}{\overset{m}{\max}}\, B_{i,j}(t)$	largest variance, greedy	$\underset{j\in J(t)}{\max}\ \underset{i\neq j}{\overset{m}{\max}}\, B_{i,j}(t) \ \le \varepsilon$

The GIFA structure is presented in Algorithm 1. `initialization` is an optional phase where all arms are sampled once. We assume that ties are randomly broken. Degrees of freedom in designing the bandit algorithm lie in the choice of the rules `compute_Jt`, `compute_bt`, `selection_rule` and `stopping_rule`, some of which may also rely on the gap indices. For example, for the stopping rule, we restrict our attention to two stopping rules already proposed for the LUCB and UGapE algorithms, respectively:

$$\tau^{LUCB} \triangleq \inf\left\{t \in \mathbb{N}^* : B_{c_t,b_t}(t) \le \varepsilon\right\} ,$$

$$\tau^{UGapE} \triangleq \inf\left\{t \in \mathbb{N}^* : \max_{j\in J(t)} \overset{m}{\underset{i\neq j}{\max}} B_{i,j}(t) \le \varepsilon\right\} .$$

3.2 Existing GIFA algorithms

We summarize in Table 1 existing and new algorithms that fit the GIFA framework.

LUCB and UGapE The only algorithms for Top-m identification in the fixed-confidence setting using adaptive sampling –namely, LUCB (Kalyanakrishnan et al., 2012) and UGapE (Gabillon et al., 2012)– were proposed for classical bandits. Both can be cast in the GIFA framework with the individual gap indices $B^{\text{ind}}_{i,j}(t) = U_i(t) - L_j(t)$ presented in Section 2. Indeed, with this choice of indices, it can be shown that the ambiguous arms $b_t = \arg\min_{j\in J(t)} L_j(t)$ and $c_t = \arg\max_{i\notin J(t)} U_i(t)$ used by both LUCB and UGapE can be rewritten as $\arg\max_{j\in J(t)} \min_{i\notin J(t)} B_{i,j}(t)$ and $\arg\max_{i\notin J(t)} B_{i,b_t}(t)$, respectively. This rewriting is crucial to propose extensions for the linear case. However, LUCB and UGapE differ by the set $J(t)$ they use and their stopping rule, as can be seen in Table 1. LUCB uses the m arms with largest empirical means for the set $J(t)$ while UGapE uses the m arms with smallest values of $\overset{m}{\underset{i\neq j}{\max}}\ B_{i,j}(t)$. This choice is motivated by Gabillon et al. (2012) by the fact that, with high probability, the quantity $\overset{m}{\underset{i\neq j}{\max}} B_{i,j}(t)$ is an upper bound on $\mu_m - \mu_j$. This is also true for more general gap indices, see Lemma 9 in Appendix D.

This observation also justifies the UGapE stopping rule: when $\max_{j\in J(t)} \overset{m}{\underset{i\neq j}{\max}}\ B_{i,j}(t) \ \le \varepsilon$, with high probability $\mu_m - \mu_j \le \varepsilon$ for all $j \in J(t)$, hence this set is likely to be included in $\mathcal{S}^{*,\varepsilon}_m$. LUCB stops when $\max_{j\in J(t)} \max_{i\notin J(t)} B_{i,j}(t) \le \varepsilon$. Interestingly, it follows from Lemma 1 below that, for fixed gap indices $(B_{i,j}(t))_{i,j\in[K]^2,t>0}$, UGapE will always stop before LUCB. Still, we chose to investigate the two stopping rules, as the sample complexity analysis of UGapE uses the upper bound $\tau^{LUCB} \geq \tau^{UGapE}$. We provide correctness guarantees for both stopping rules in Section 4.1 and investigate their practical impact in Section 5.

Lemma 1. *For all $t > 0$, for any subset J of size m, for all $j \in J$, $\overset{m}{\underset{i\neq j}{\max}} B_{i,j}(t) \le \max_{i\notin J} B_{i,j}(t)$.*

(proof in Appendix D)

Regarding the selection rule, UGapE selects the least sampled arm among b_t and c_t, which coincides with the largest variance rule that we propose for the general linear setting:

- "largest variance": $\arg\max_{a\in\{b_t,c_t\}} W_t(a)$.

In the original version of LUCB, both arms b_t and c_t are sampled at time t, but the analysis that we propose in this paper obtains similar guarantees for LUCB using the largest variance rule, so we only consider this selection rule in the remainder of the paper.

LinGapE A striking example of adaptive sampling in linear BAI (thus, only for $m = 1$) is LinGapE (Xu et al., 2018). The first novelty in LinGapE compared to UGapE and LUCB is the use of paired gap indices $B^{\text{pair}}_{i,j}(t)$ exploiting the arm features. Paired indices may indeed increase performance. First, it follows from the triangular inequality for the Mahalanobis norm $\| \cdot \|_{\hat{\Sigma}_t^\lambda}$ that $||x_i - x_j||_{\hat{\Sigma}_t^\lambda} \leq ||x_i||_{\hat{\Sigma}_t^\lambda} + ||x_j||_{\hat{\Sigma}_t^\lambda}$ for all pairs of arms (i, j) ; therefore, if both types of gap indices use the same threshold $C_{t,\delta}$, paired indices are smaller: $B^{\text{pair}}_{i,j}(t) \leq B^{\text{ind}}_{i,j}(t)$. Moreover, Lemma 2 below (proved in Appendix D) implies that paired or individual indices using arm features always yield smaller bounds on the gaps than individual indices without arm features, that take the form

$$\hat{\Delta}_{i,j}(t) + C_{t,\delta}\left[\frac{1}{\sqrt{N_i(t)}} + \frac{1}{\sqrt{N_j(t)}}\right] .$$

Lemma 2. $\forall t > 0, \forall a \in [K], \forall y \in \mathbb{R}^N$,

$$||y||_{(\hat{V}_t^\lambda)^{-1}} \leq ||y||/\left(\sqrt{N_a(t)||x_a||^2 + \lambda}\right) .$$

The second novelty in LinGapE is the proposed selection rules. Let X be the matrix which columns are the arm contexts, and $\|\cdot\|_1 : v \mapsto \sum_i |v_i|$. Instead of selecting either b_t or c_t, Xu et al. (2018) propose two selection rules to possibly sample another arm that would reduce the variance of the estimate $\theta^\top(x_{b_t} - x_{c_t})$:

- “greedy”: $\arg\min_{a\in[K]} ||x_{b_t} - x_{c_t}||_{(\hat{V}^\lambda_{t-1} + x_a x_a^\top)^{-1}}$,
- “optimized”: if $w^*(b_t, c_t) = \underset{w\in\mathbb{R}^K : x_{b_t} - x_{c_t} = wX^T}{\arg\min} \|w\|_1$,

$$\underset{a\in[K]: w^*_a(b_t,c_t)>0}{\arg\max} N_a(t)\frac{||w^*(b_t, c_t)||_1}{|w^*_a(b_t, c_t)|}. \qquad (1)$$

As explained by Xu et al. (2018), these two rules are meant to bring the empirical proportions of selections close to an optimal design $(\lambda_1, \ldots, \lambda_K)$ asymptotically minimizing $\|x_{b_t} - x_{c_t}\|_{(\sum_a \lambda_a x_a x_a^\top)^{-1}}$.

3.3 m-LinGapE and LinGIFA

Building on the interesting features of existing algorithms, we now propose two new algorithms for Top-m identification in linear bandits, called m-LinGapE and LinGIFA and described in Table 1.

m-LinGapE is an extension of LinGapE (Xu et al., 2018) to linear Top-m, which coincides with the original algorithm for $m = 1$. It may also be viewed as an extension of LUCB using paired indices. We investigate three possible selection rules for this algorithm: largest variance, greedy and optimized. LinGIFA is inspired by UGapE (Gabillon et al., 2012), and also uses paired gap indices. Note that LinGIFA has a unique `compute_bt` rule $b_t = \arg\max_{j\in J(t)} \overset{m}{\underset{i\neq j}{\max}} B_{i,j}(t)$, which does not coincide with any of the previously mentioned rules. In Table 1, we considered the stopping rules of the original algorithms they derive from. As LinGapE is one of the most performant algorithms for linear BAI (see for instance experiments in Degenne et al. (2020)), we expect m-LinGapE to work well for Top-m. However, the strength of LinGIFA is that it is completely defined in terms of gap indices, and, as such, can easily be tuned for performance by deriving tighter bounds on the gaps. Moreover, LinGIFA relies on a stopping rule that is more aggressive than that used by m-LinGapE, as discussed above.

We emphasize that provided that the regularizing constant λ in the design matrix is positive, both $m-$LinGapE and LinGIFA can be run without an initialization phase (as done in the experiments for LinGIFA). This permits to avoid the initial sampling cost when the number of arms is large that was noticed by Fiez et al. (2019).

4 THEORETICAL GUARANTEES

In this section, we present an analysis of m-LinGapE and LinGIFA. The fact that these algorithms are (ε, m, δ)-PAC is a consequence of generic correctness guarantees that can be obtained for GIFA algorithms (even not fully specified) and is presented in Section 4.1. We further analyze in Section 4.2 the sample complexity of LUCB-like algorithms, which comprise m-LinGapE.

4.1 Correctness of GIFA instances

We justify that the two stopping rules introduced above lead to (ε, m, δ)-PAC algorithms, provided a condition on the gap indices given in Definition 1.

Definition 1 (Good gap indices)**.** Let us denote

$$\mathcal{E}_m^{GIFA} \triangleq \bigcap_{t>0} \bigcap_{j\in(\mathcal{S}_m^{*,\varepsilon})^c} \bigcap_{k\in\mathcal{S}_m^\star} \left(B_{k,j}(t) \geq \mu_k - \mu_j\right) .$$

A *good choice* of gap indices $(B_{i,j}(t))_{i,j\in[K],t>0}$ satisfies $\mathbb{P}(\mathcal{E}_m^{GIFA}) \geq 1 - \delta$.

First, we observe that on the event $\mathcal{E}_m^{GIFA}$ introduced in Definition 1 both stopping rules τ^{UGapE} and τ^{LUCB} output an (ε, δ) correct answer.

Theorem 1. *On the event* $\mathcal{E}_m^{GIFA}$*:* **(i)** *any GIFA algorithm using* $b_t = \arg\max_{j\in J(t)} \max_{i\notin J(t)} B_{i,j}(t)$ *sat-*

isfies $J(\tau^{LUCB}) \subseteq \mathcal{S}_m^{*,\varepsilon}$ and **(ii)** *any GIFA algorithm using $b_t \in J(t)$ satisfies $J(\tau^{UGapE}) \subseteq \mathcal{S}_m^{*,\varepsilon}$.*

Proof. Let us assume by contradiction that there is an arm $b \in J(t) \cap (\mathcal{S}_m^{*,\varepsilon})^c$ at stopping time t. We first prove that in both cases there exists $c \in \mathcal{S}_m^\star$ such that $B_{c,b}(t) \leq \varepsilon$ (*). Assuming that (*) does not hold, observe that $\mathcal{S}_m^\star \subseteq \{a \neq b, B_{a,b}(t) > \varepsilon\}$ **(a)** and then $\arg\overset{m}{\max}_{a \neq b} B_{a,b}(t) \in \{a \neq b, B_{a,b}(t) > \varepsilon\}$ **(b)** since $b \notin \mathcal{S}_m^\star$ and $|\mathcal{S}_m^\star| = m$. Depending on the definition of b_t and on the stopping rule, we now split the proof into two parts:

(i). Using the definition of τ^{LUCB}, c_t and b_t, it holds that $\forall b \in J(t), \forall c \notin J(t), B_{c,b}(t) \leq \varepsilon$. Then, using **(a)**, $J(t)^c \subseteq \{a \neq b, B_{a,b}(t) \leq \varepsilon\}$, which means $\mathcal{S}_m^\star \cap (J(t))^c = \emptyset$. Thus $\mathcal{S}_m^\star = J(t)$ (because $|\mathcal{S}_m^\star| = |J(t)| = m$) whereas $b \in J(t) \cap (\mathcal{S}_m^\star)^c$, which is a contradiction. Hence (*) holds.

(ii). Using the definition of τ^{UGapE}, for any $b \in J(t)$, $\overset{m}{\max}_{a \neq b}\ B_{a,b}(t) \leq \epsilon$ holds. However, **(b)** means $\overset{m}{\max}_{a \neq b}\ B_{a,b}(t) > \epsilon$, which is a contradiction: (*) holds.

Then, at stopping time t, there exists $b \in J(t) \cap (\mathcal{S}_m^{*,\varepsilon})^c$ and $c \in \mathcal{S}_m^\star$ such that $B_{c,b}(t) \leq \varepsilon$. Using successively the definition of the event $\mathcal{E}_m^{GIFA}$, and the fact that $c \in \mathcal{S}_m^\star$, this means that there exists $b \in J(t)^c \cap (\mathcal{S}_m^\star)^c$ such that $\mu_b \geq \mu_m - \varepsilon$. Then $b \in (\mathcal{S}_m^{*,\varepsilon})^c \cap \mathcal{S}_m^{*,\varepsilon}$, which is absurd. Hence this proves that $J(t) \subseteq \mathcal{S}_m^{*,\varepsilon}$ on the event $\mathcal{E}_m^{GIFA}$ at stopping time t. □

It easily follows from Theorem 1 that if m-LinGapE or LinGIFA are based on good gap indices in the sense of Definition 1, both algorithms are (ε, m, δ)-PAC. We exhibit below a threshold for which the corresponding paired indices $B_{i,j}^{\text{pair}}(t)$ and individual indices $B_{i,j}^{\text{ind}}(t)$ (defined in Section 2) are good gap indices.

Lemma 3. *For indices of the form $B_{i,j}(t) = \hat{\Delta}_{i,j}(t) + C_{\delta,t} W_t(i,j)$ with $W_t(i,j) = \|x_i - x_j\|_{\hat{\Sigma}_t^\lambda}$ (paired) or $W_t(i,j) = ||x_i||_{\hat{\Sigma}_t^\lambda} + ||x_j||_{\hat{\Sigma}_t^\lambda}$ (individual) with*

$$C_{\delta,t} = \sqrt{2\ln\left(\frac{1}{\delta}\right) + N \ln\left(1 + \frac{(t+1)L^2}{\lambda^2 N}\right)} + \frac{\sqrt{\lambda}}{\sigma} S, \tag{2}$$

where we recall that $\max_{a \in [K]} \|x_a\| \leq L$ and $\|\theta\| \leq S$, we have $\mathbb{P}(\mathcal{E}_m^{GIFA}) \geq 1 - \delta$.

Proof. The proof follows from the fact that

$$\left\{\forall t \in \mathbb{N}^* : \|\hat{\theta}_t^\lambda - \theta\|_{(\hat{\Sigma}_t^\lambda)^{-1}} \leq C_{\delta,t}\right\} \subseteq \mathcal{E}_m^{GIFA}$$

together with Lemma 4.1 in Kaufmann (2014) which yields $\mathbb{P}\left(\forall t \in \mathbb{N}^* : \|\hat{\theta}_t^\lambda - \theta\|_{(\hat{\Sigma}_t^\lambda)^{-1}} \leq C_{\delta,t}\right) \geq 1 - \delta$. For paired indices, the inclusion follows from the fact that

$$|(\hat{\mu}_i(t) - \hat{\mu}_j(t)) - (\mu_i - \mu_j)| \leq \|\hat{\theta}_t^\lambda - \theta\|_{(\hat{\Sigma}_t^\lambda)^{-1}} \|x_i - x_j\|_{\hat{\Sigma}_t^\lambda},$$

where we use that $x^\top y \leq \|x\|_{\Sigma^{-1}} \|y\|_\Sigma$ for any $x, y \in \mathbb{R}^N$ and positive definite $\Sigma \in \mathbb{R}^{N \times N}$. For individual indices, we further need the triangular inequality $\|x_i - x_j\|_{\hat{\Sigma}_t^\lambda} \leq \|x_i\|_{\hat{\Sigma}_t^\lambda} + \|x_j\|_{\hat{\Sigma}_t^\lambda}$ to prove the inclusion. □

4.2 Sample complexity results

We derive below a high-probability upper bound on the sample complexity of a subclass of GIFA algorithms which comprise m-LinGapE, combined with different selection rules. More precisely, we upper bound the sample complexity on

$$\mathcal{E} \triangleq \bigcap_{t>0} \bigcap_{i,j \in [K]} \Big(\mu_i - \mu_j \in [-B_{j,i}(t), B_{i,j}(t)]\Big),$$

an event which is trivially included in $\mathcal{E}^{\text{GIFA}}$ and which holds therefore with probability larger than $1 - \delta$ with the choice of threshold (2). To state our results, we define the true gap of an arm k as $\Delta_k \triangleq \mu_k - \mu_{m+1}$ if $k \in \mathcal{S}_m^\star$, $\mu_m - \mu_k$ otherwise ($\Delta_k \geq 0$ for any $k \in [K]$).

Theorem 2. *For m-LinGapE, on event $\mathcal{E}$ on which algorithm $\mathcal{A}$ is (ε, m, δ)-PAC, stopping time τ_δ satisfies $\tau_\delta \leq \inf\{u \in \mathbb{R}^{*+} : u > 1 + H^\varepsilon(\mathcal{A}, \mu) C_{\delta,u}^2 + \mathcal{O}(K)\}$, where:*

(i). *for $\mathcal{A} = m$-LinGapE with the largest variance selection rule*[3] *:*

$$H^\varepsilon(\mathcal{A}, \mu) \triangleq 4\sigma^2 \sum_{a \in [K]} \max\left(\varepsilon, \frac{\varepsilon + \Delta_a}{3}\right)^{-2},$$

(ii). *for $\mathcal{A} = m$-LinGapE with the optimized selection rule in Xu et al. (2018) :*

$$H^\varepsilon(\mathcal{A}, \mu) \triangleq \sigma^2 \sum_{a \in [K]} \max_{i,j \in [K]} \frac{|w_a^*(i,j)|}{\max\left(\varepsilon, \frac{\varepsilon + \Delta_i}{3}, \frac{\varepsilon + \Delta_j}{3}\right)^2},$$

where $w^*(i,j)$ satisfies Equation 1 applied to the arm pair (i,j).

Upper bounds for all analyzed algorithms and classical counterparts are shown in Table 2.

4.3 Sketch of proof

The proof of Theorem 2 generalizes and extends the proofs for classical Top-m and linear BAI (Xu et al., 2018), with paired or individual gap indices. We sketch below the sample complexity analysis of a specific subclass of GIFA algorithms, postponing the proof of some

[3] or pulling both arms in $\{b_t, c_t\}$ at time t

Table 2: Sample complexity results for linear and classical Top-m algorithms. w^* is defined as in Equation (1). The additive term in K is due to the initialization phase. Our proposal is in bold type.

Algorithm	**Complexity constant $\mathrm{H}^\varepsilon(\cdot, \mu)$**	**Upper bound on τ_δ**
LUCB	$2 \sum_{a \in [K]} \max\left(\frac{\varepsilon}{2}, \Delta_a\right)^{-2}$	$\inf_{u>0}\left\{u > 1 + \mathrm{H}^\varepsilon(\mathrm{LUCB}, \mu) C^2_{\delta,u} + K\right\}$
UGapE	$2 \sum_{a \in [K]} \max\left(\varepsilon, \frac{\varepsilon + \Delta_a}{2}\right)^{-2}$	$\inf_{u>0}\left\{u > 1 + \mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu) C^2_{\delta,u} + K\right\}$
***m*-LinGapE (1)** **(largest variance)**	$4\sigma^2 \sum_{a \in [K]} \max\left(\varepsilon, \frac{\varepsilon + \Delta_a}{3}\right)^{-2}$	$\inf_{u>0}\left\{u > 1 + \mathrm{H}^\varepsilon(m\text{-LinGapE}(1), \mu) C^2_{\delta,u}\right\}$
***m*-LinGapE (2)** **(optimized)**	$\sigma^2 \sum_{a \in [K]} \max_{i,j \in [K]} \frac{\lvert w^*_a(i,j)\rvert}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_i}{3}, \frac{\varepsilon+\Delta_j}{3}\right)^2}$	$\inf_{u>0}\left\{u > 1 + \mathrm{H}^\varepsilon(m\text{-LinGapE}(2), \mu) C^2_{\delta,u}\right\}$

auxiliary lemmas to Appendix C. We focus on GIFA algorithms that use

$$J(t) \triangleq \operatorname*{arg\,max}^{[m]}_{i \in [K]} \hat{\mu}_i(t) \text{ and } b_t \triangleq \operatorname*{arg\,max}_{j \in J(t)} \max_{i \notin J(t)} B_{i,j}(t) \,.$$

This includes LinGapE, m-LinGapE, but also LUCB. The key ingredient in the proof is the following lemma, which holds for any gap indices of the form $B_{i,j}(t) \triangleq \hat{\mu}_i(t) - \hat{\mu}_j(t) + W_t(i,j)$ for $i, j \in [K]^2$.

Lemma 4. *On the event $\mathcal{E}$, for all $t > 0$,*

$$B_{c_t,b_t}(t) \leq \min(-(\Delta_{b_t} \vee \Delta_{c_t}) + 2W_t(b_t, c_t), 0) + W_t(b_t, c_t).$$

This result is a counterpart of Lemma 4 in Xu et al. (2018), but does not require $|J(t)| = 1$ at a given time $t > 0$, notably by noticing that, by definition of b_t and c_t, $B_{c_t,b_t}(t) = \max_{j \in J(t)} \max_{i \notin J(t)} B_{i,j}(t)$. In order to get the upper bound in Theorem 2 for m-LinGapE using the optimized rule, one can straightforwardly apply Lemma 1 in Xu et al. (2018) to the inequality stemming from Lemma 4. For the selection rules which either select both b_t and c_t, or the arm in $\{b_t, c_t\}$ with the largest variance term, by combining Lemma 4 with the definition of the stopping rule τ^{LUCB}, we obtain the following upper bound on $N_{a_t}(t)$, where a_t is (one of) the pulled arm(s) at time $t < \tau_\delta$.

Lemma 5. $\forall t > 0, \tau_\delta > t, N_{a_t}(t) \leq T^*(a_t, \delta, t)$*, where a_t is a pulled arm at time t, and*

$$T^*(a_t, \delta, t) = 4\sigma^2 C^2_{\delta,t} \max\left(\varepsilon, \frac{\varepsilon + \Delta_{a_t}}{3}\right)^{-2}.$$

Finally, the upper bound in Theorem 2 is a consequence of the following result.

Lemma 6. *Let $T^* : [K] \times (0,1) \times \mathbb{N}^* \to \mathbb{R}^{*+}$ be a function that is nondecreasing in t, and $\mathcal{I}_t$ the set of pulled arms at time t. Let $\mathcal{E}$ be an event such that for all $t < \tau_\delta, \delta \in (0,1)$, $\exists a_t \in \mathcal{I}_t, N_{a_t}(t) \leq T^*(a_t, \delta, t)$. Then it holds on the event $\mathcal{E}$ that $\tau_\delta \leq T(\mu, \delta)$ where*

$$T(\mu, \delta) \triangleq \inf\left\{u \in \mathbb{R}^{*+} : u > 1 + \sum_{a=1}^{K} T^*(a, \delta, u)\right\}.$$

4.4 Discussion

If $\mathrm{H}^\varepsilon(\mathcal{A}, \mu)$ is the complexity constant associated with bandit algorithm $\mathcal{A}$ and bandit instance $\mu \in \mathbb{R}^K$, then, by looking at Table 2, we can first notice that

$$\mathrm{H}^\varepsilon(\mathrm{LUCB}, \mu) \geq \mathrm{H}^\varepsilon(m\text{-LinGapE}(1), \mu) \geq \mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu).$$

Note that the quantity $w^*_a(i,j)$ featured in the complexity quantity associated to m-LinGapE with the optimized selection rule (m-LinGapE(2)) is not *a priori* bounded by a constant – see Equation (1). Authors in (Xu et al., 2018) have shown that in specific instances, $\mathrm{H}^\varepsilon(m\text{-LinGapE}(2), \mu) \leq \frac{9}{8}\mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu)$. To further compare these two complexity quantities, we designed the following experiment to estimate how many times $\mathrm{H}^\varepsilon(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu)$: $K \times N$ values are randomly sampled from a Gaussian distribution $N(0, D)$. From these we build matrix $X' \in \mathbb{R}^{N \times K}$, which is then column-normalized (norm $\|\cdot\|$) to yield feature matrix X. We use $\theta = e_1$, where $e_1 = [1, 0, \dots, 0]^\top \in \mathbb{R}^K$ and $\|\theta\| = 1$. Then we compute both constants $\mathrm{H}^\varepsilon(m\text{-LinGapE}(2), \mu)$ and $\mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu)$ where $\mu = X\theta$ and $m = \left[\frac{K}{3}\right]$, check whether $\mathrm{H}^\varepsilon(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^\varepsilon(\mathrm{UGapE}, \mu)$. We perform this experiment $1,000$ times and report in Table 4 (in Appendix A) the fraction of times this condition holds, for multiple values of K (number of arms), N (dimension) and D (variance of the Gaussian distribution). We observe that, in these artificially generated instances, the condition $\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^{\varepsilon}(\mathrm{UGapE}, \mu)$ is seldom verified.

Hence, our theoretical analysis suggests that, in these cases –which usually turn out to be really hard linear instances– LinGIFA, which structure is similar to that of UGapE, may perform better than m-LinGapE. However, as we will see below, the practical story is different and m-LinGapE with the optimized selection rule can be as efficient, and often better than LinGIFA using the "largest variance" selection rule.

5 EXPERIMENTAL STUDY

In this section, we report the results of experiments on simulated data and on a simple drug repurposing instance. In all experiments, we use $\sigma = 1/2$, $\varepsilon = 0$, $\delta = 5\%$. For all artificially generated experimental settings, we use Gaussian arms, that is, at time t, $r_t \sim N(\mu_{a_t}, \sigma^2)$. For the tuning of our algorithm we set the regularization parameter to $\lambda = \sigma/20$, following Hoffman et al. (2014). In an effort to respect the trade-off between identification performance and speed, we set $C_{\delta,t} = \sqrt{2\ln((\ln(t)+1)/\delta)}$ ("heuristic" value) if the empirical error remained below δ instead of the theoretically valid threshold (2) that guarantees all algorithms to be $(\varepsilon, m, \delta)-$PAC. Otherwise, we use (2), and for classical bandits, we use the threshold used in Kalyanakrishnan et al. (2012). We consider the following experimental settings.

Classic instances We consider K arms such that, for $a \in [m]$, $x_a = e_1 + e_a$, $x_{m+1} = \cos(\omega)e_1 + \sin(\omega)e_{m+1}$, and for any $a > m+1$, $x_a = e_{a-1}$, and $\theta = e_1$, where $(e_i)_{i\in[K-1]}$ are the canonical basis of $\mathbb{R}^{K-1}$. This construction extends the usual "hard" instance for BAI in linear bandits ($m = 1$), originally proposed by Soare et al. (2014) where one considers three arms, of respective feature vectors $x_1 = [1,0]^\top, x_2 = [0,1]^\top$ and $x_3 = [\cos(\omega), \sin(\omega)]^\top$, and $\theta = e_1$, where $\omega \in (0, \frac{\pi}{2})$. Arm 1 is the best arm, and as ω decreases, it becomes harder to discriminate between arms 1 and 3, and as such, pulling suboptimal arm 2 might be useful.

Drug repurposing instance Our drug-scoring function relies on the simulation of the treatment effect on gene activity via a Boolean network (Kauffman, 1969). The details are laid out in Appendix B. When a drug is selected, a reward is generated by applying the simulator to the gene activity profile of a patient chosen at random, and computing the cosine score between the post-treatment gene activity profile and the healthy gene activity profile. The higher this score is, the most similar the final treated patient and healthy samples are. We consider as drug features the treated gene activity profiles, which belong to $\mathbb{R}^{71}$.

Results Each figure reports the empirical distribution of the sample complexity (estimated over 500 runs) for different algorithms. In all the experiments, the empirical error is always below 5%, and is reported in Table 3.

Figure 1(a) reports the results for a classic problem with $m = 2, K = 4$ and $N = 3$. We compare two algorithms that are not using the arm features, LUCB and UGapE, with GIFA algorithms based on contextual indices. We investigate the use of individual and paired gap indices, different selection rules, and the two stopping rules τ^{LUCB} and τ^{UGapE} in both m-LinGapE and LinGIFA. We observe that there is almost no difference in the use of either stopping rules, but using paired indices leads to noticeably better sample complexity than individual ones. As noticed in Xu et al. (2018) for $m = 1$, m-LinGapE with the optimized and the greedy sampling rules have similar performance. In addition, using the greedy or optimized rules leads to slightly better performance compared to the largest variance rule. More importantly, we observe that linear bandit algorithms largely outperform their classical counterparts, even on a rather easy instance ($\omega = \frac{\pi}{6}$, hence $\mu_m - \mu_{m+1} \approx 0.13$).

Figure 1(b) shows the empirical sample complexity on a hard Top-1 classic problem ($\omega = 0.1$, $\mu_1 - \mu_2 \approx 0.005$) which allows to compare our algorithms to the state-of-the-art LinGame algorithm. $m-$LinGapE and LinGIFA have a better performance than LinGame, which is encouraging, even if our algorithms do not have any optimality guarantees in theory.

Figure 1(c) displays the result for a drug repurposing instance for epilepsy, including $K = 10$ arms (drugs) with 5 anti-epileptic and 5 pro-convulsant drugs, hence the choice $m = 5$ (a close-up is shown in Appendix B without LUCB). Although the linear dependency between rewards and features may be violated in this real-world example, taking into account features still helps in considerably reducing the sample complexity (approximately by a factor $\frac{1}{2}$).

6 DISCUSSION

To our knowledge, we have provided the first unified framework and fully adaptive algorithms for linear Top-m. Our theoretical analysis shows our algorithms do not perform any worse than their clas-

Table 3: Error frequencies rounded to 5 decimal places, for each Top-m and BAI algorithm (averaged across 500 runs). Proposed algorithms' names are in bold type. Each column corresponds to one figure.

Algorithm	(1)a	(1)b	(1)c
m-LinGapE (greedy)	0.0	0.044	0.0
m-LinGapE (optimized)	0.0	-	0.0
LinGIFA (largest var.)	0.0	-	0.0
LinGIFA (τ^{LUCB}, largest var.)	0.0	-	-
LinGIFA (τ^{LUCB}, greedy)	0.0	-	-
LinGIFA (greedy)	0.0	0.0	0.0
LinGIFA (individual indices)	0.0	-	-
LUCB (largest var.)	0.0	-	0.0
LUCB (sampling both arms)	-	-	0.0
UGapE	0.0	-	-
LinGame	-	0.0	-

sical counterparts. Code is publicly available at `https://github.com/clreda/linear-top-m`. However, in real life and in our drug repurposing instance, the linear dependency between features and rewards does not hold. A future direction of our work would be dealing with such model misspecification. Another perspective would be the analysis of the greedy sampling rule. Indeed, this sampling rule leads to more efficient algorithms in our linear experiments.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their feedback, Xuedong Shang for his valuable input for the implementation of LinGame, and Baptiste Porte for his help on building the drug repurposing instance applied to epilepsy. This work was supported by the Institut National pour la Santé et la Recherche Médicale (INSERM, France), the Centre National de la Recherche Scientifique (CNRS, France), the Université Sorbonne Paris Nord, the Université de Paris, France, the French Ministry of Higher Education and Research [ENS.X19RDTME-SACLAY19-22] (C. R.), the French National Research Agency [ANR-18-CE17-0009-01 and ANR-18-CE37-0002-03] (A. D.-D.), [ANR-19-CE23-0026-04] (E. K.), and by the "Digital health challenge" Inserm CNRS joint program (C. R., E. K. and A. D.-D.).

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3(null): 397–422, 2003. ISSN 1532-4435.

Adam S Brown and Chirag J Patel. A standard

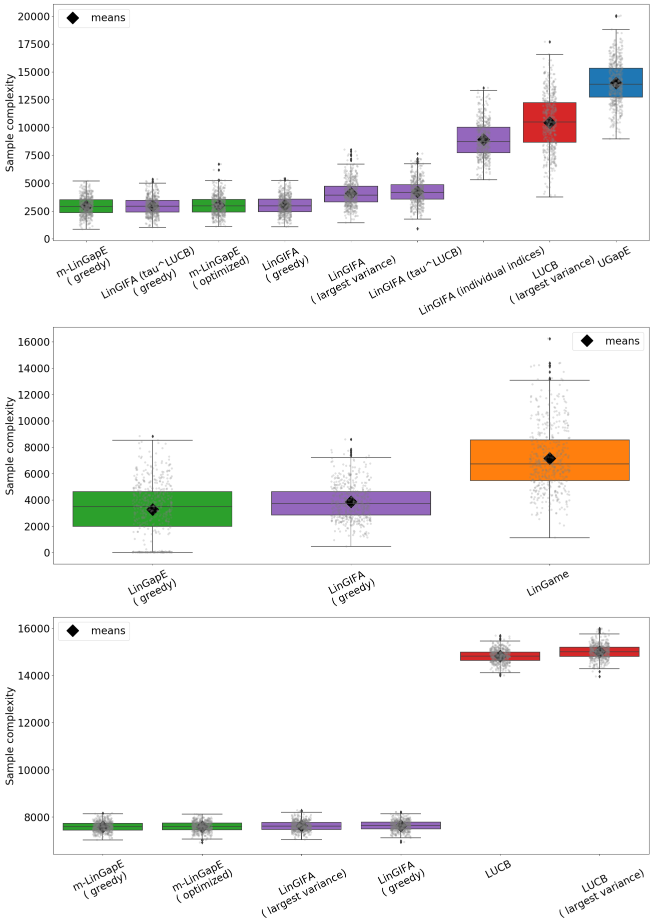

Figure 1: From top to bottom: classic instances (a) $K = 4, \omega = \frac{\pi}{6}, m = 2$; (b) $K = 3, \omega = 0.1, m = 1$; (c) drug repurposing instance $K = 10$, $m = 5$. Lines are quantiles and jittered individual outcomes are plotted in grey.

database for drug repositioning. *Scientific data*, 4: 170029, 2017.

Séastian Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Fondations and Trends in Machine Learning*, 5(1):1–122, 2012.

Sébastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 258–265, Atlanta, Georgia, USA, 2013. PMLR. URL http://proceedings.mlr.press/v28/bubeck13.html.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, ALT'09, page

23–37, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 3642044131.

Lijie Chen, Jian Li, and Mingda Qiao. Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 101–110, Fort Lauderdale, FL, USA, 2017. PMLR. URL http://proceedings.mlr.press/v54/chen17a.html.

Lijun Cheng and Lang Li. Systematic quality control analysis of lincs data. *CPT: pharmacometrics & systems pharmacology*, 5(11):588–598, 2016.

Neil R Clark, Kevin S Hu, Axel S Feldmann, Yan Kou, Edward Y Chen, Qiaonan Duan, and Avi Ma'ayan. The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC bioinformatics*, 15(1):79, 2014.

Rémy Degenne and Wouter M Koolen. Pure exploration with multiple correct answers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 14591–14600. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/60cb558c40e4f18479664069d9642d5a-Paper.pdf.

Rémy Degenne, Pierre Ménard, Xuedong Shang, and Michał Valko. Gamification of pure exploration for linear bandits. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2432–2442. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/degenne20a.html.

Andrée Delahaye-Duriez, Prashant Srivastava, Kirill Shkura, Sarah R Langley, Liisi Laaniste, Aida Moreno-Moral, Bénédicte Danis, Manuela Mazzuferi, Patrik Foerch, Elena V Gazina, et al. Rare and common epilepsies converge on a shared gene regulatory network providing opportunities for novel antiepileptic drug discovery. *Genome biology*, 17(1): 1–18, 2016.

Qiaonan Duan, St Patrick Reid, Neil R Clark, Zichen Wang, Nicolas F Fernandez, Andrew D Rouillard, Ben Readhead, Sarah R Tritsch, Rachel Hodos, Marc Hafner, et al. L1000cds 2: Lincs l1000 characteristic direction signatures search engine. *NPJ systems biology and applications*, 2:16015, 2016.

Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 10667–10677. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/8ba6c657b03fc7c8dd4dff8e45defcd2-Paper.pdf.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 3212–3220. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/8b0d268963dd0cfb808aac48a549829f-Paper.pdf.

Matthew Hoffman, Bobak Shahriari, and Nando Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 365–374, Reykjavik, Iceland, 2014. PMLR. URL http://proceedings.mlr.press/v33/hoffman14.html.

Thomas J Hwang, Daniel Carpenter, Julie C Lauffenburger, Bo Wang, Jessica M Franklin, and Aaron S Kesselheim. Failure of investigational drugs in late-stage clinical development and publication of trial results. *JAMA internal medicine*, 176(12):1826–1833, 2016.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 227–234, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

Émilie Kaufmann. *Analyse de stratégies bayésiennes et fréquentistes pour l'allocation séquentielle de ressources.* PhD thesis, 2014. URL http://www.theses.fr/2014ENST0056. Thèse de doctorat dirigée par Cappé, Olivier et Garivier, Aurélien, Signal et images Paris, ENST 2014.

Émilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 228–251, Princeton, NJ, USA, 2013. PMLR. URL http://proceedings.mlr.press/v30/Kaufmann13.html.

Aliyu Musa, Laleh Soltan Ghoraie, Shu-Dong Zhang, Galina Glazko, Olli Yli-Harja, Matthias Dehmer, Benjamin Haibe-Kains, and Frank Emmert-Streib. A review of connectivity map and computational approaches in pharmacogenomics. *Briefings in bioinformatics*, 19(3):506–523, 2018.

Clémence Réda and Wilczyński. Automated inference of gene regulatory networks using explicit regulatory modules. *Journal of Theoretical Biology*, 486:110091, 2020. ISSN 0022-5193. doi: https://doi.org/10.1016/j.jtbi.2019.110091. URL http://www.sciencedirect.com/science/article/pii/S0022519319304606.

Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 828–836. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/f387624df552cea2f369918c5e1e12bc-Paper.pdf.

Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452, 2017.

Damian Szklarczyk, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, et al. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, page gkw937, 2016.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 843–851. PMLR, 2018. URL http://proceedings.mlr.press/v84/xu18d.html.

Boyan Yordanov, Sara-Jane Dunn, Hillel Kugler, Austin Smith, Graziano Martello, and Stephen Emmott. A method to identify and analyze biological programs through automated reasoning. *NPJ systems biology and applications*, 2:16010, 2016.

A COMPARING COMPLEXITY CONSTANTS

A.1 About the classic instance for linear Top-m identification

In higher dimensions, and when $m = K - 2$, ω can be seen as the angle between the $m + 1$-best arm vector and the hyperplane formed by the m best arm feature vectors. In order to check if, for $m \geq 1$, decreasing the value of $\omega \in (0, \frac{\pi}{2})$ yields to harder instances (as it is for $m = 1$), we ran the bandit algorithms on the instance $K = 4$, $N = 3$, $m = 2$ for $\omega \in \{\frac{\pi}{3}, \frac{\pi}{6}\}$. The resulting boxplots are shown in Figure 2. It can then be seen that indeed, for all algorithms, the empirical average sample complexity increases as ω decreases, which is an argument in favour of the use of this type of instance for the test of linear Top-m algorithms.

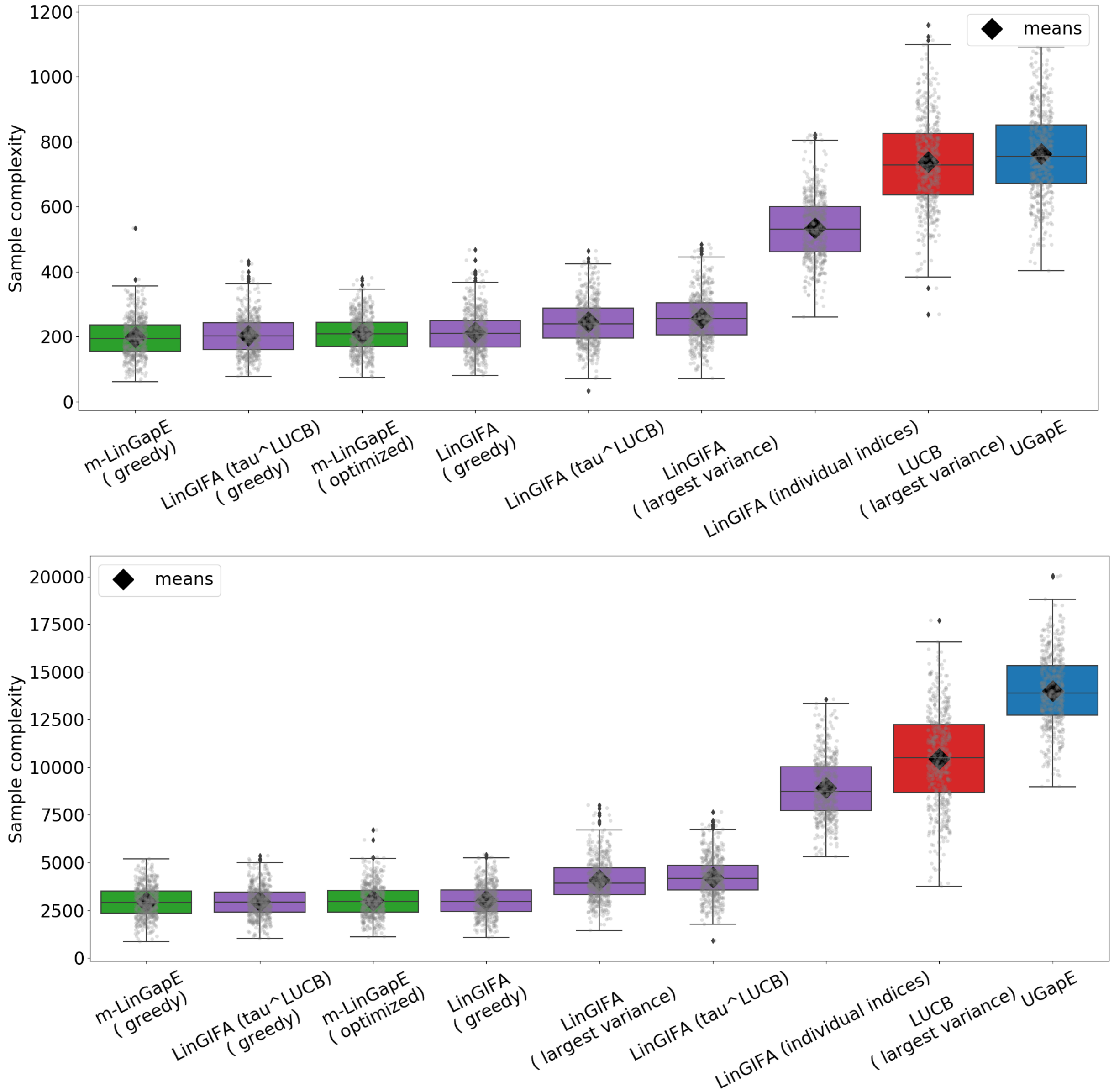

Figure 2: From top to bottom: classic instances (a) $K = 4$, $\omega = \frac{\pi}{3}$, $m = 2$; (b) $K = 4$, $\omega = \frac{\pi}{6}$, $m = 2$. Error frequencies are rounded up to 5 decimal places.

A.2 Comparing complexity constants

Below is the table to which we refer to in Section 4.4.

Table 4: Comparison of complexity constants in m-LinGapE and UGapE (% on $1,000$ random instances).

D	0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.5
K	10	10	10	10	10	10	20	20	20	20	20	30	30	30	30
m	4	4	4	4	4	4	7	7	7	7	7	11	11	11	11
N	5	5	10	10	20	20	10	20	20	40	40	15	15	30	30
%	29.1%	30.8%	0.0%	0.0%	0.0%	0.0%	0.6%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%

We have tested if, empirically, LinGIFA was more performant than LinGapE on instances where $\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^{\varepsilon}(\text{UGapE}, \mu)$, since LinGIFA has a similar structure as UGapE. We generated a random linear instance, following the procedure described in Section 4.4 in the paper, with $K = 10$, $N = 5$, $D = 0.5$. For $m = 3$, the condition $\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^{\varepsilon}(\text{UGapE}, \mu)$ is satisfied, whereas it is not when $m = 8$. We considered Gaussian reward distributions. See Figure 3. From these results, we notice that both algorithms are actually similar in sample complexity in both instances. Hence, even if the condition $\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^{\varepsilon}(\text{UGapE}, \mu)$ is seldom satisfied as seen in Table 4, in practice, m-LinGapE with the optimized rule is still performant.

Table 5: Values of complexity constants in m-LinGapE and UGapE on the randomly generated instance.

	$m = 3$	$m = 8$
$\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu)$	$4,545.97$	$32,124.01$
$\mathrm{H}^{\varepsilon}(\text{UGapE}, \mu)$	$5,047.76$	$27,622.18$
$\mu_m - \mu_{m+1}$	0.075	0.029
$\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(2), \mu) \leq \mathrm{H}^{\varepsilon}(\text{UGapE}, \mu)$?	**True**	**False**

B DRUG REPURPOSING INSTANCE

Remember that we call "phenotypes" gene activity profiles of patients and controls (healthy group) (that is, vectors which represent the genewise activity in a finite set of genes). We focus on a finite set of genes called M30, which has been shown to have a global gene activity that is anti-correlated to epileptic gene activity profiles (Delahaye-Duriez et al., 2016). The bandit instance comprises of arms/drugs, which, once pulled, return a single score/reward which quantifies their ability to "reverse" a epileptic phenotype –that is, an anti-epileptic treated epileptic phenotype should be closer to a healthy phenotype. A gene regulatory network (GRN) is a summary of gene transcriptomic interactions as a graph: nodes are genes or proteins, (directed) edges are regulatory interactions. We see a GRN as a Boolean network (BN): nodes can have two states (0 or 1), and each of them is assigned a so-called "regulatory function", that is, a logical formulæ which updates their state given the states of regulators (i.e., predecessors in the network) at each time step. In order to infer the effect of a treatment, one can set as initial network state ("initial condition") the patient phenotype masked by the perturbations on the drug targets, and iteratively update the network state until reaching an attractor state (`phenotype_prediction` procedure).

Building the Boolean network We use the Boolean network inference method described in (Yordanov et al., 2016) using code at repository `https://github.com/regulomics/expansion-network` (Réda and Wilczyński, 2020). We get the unsigned undirected regulatory interactions from the *protein-protein interaction network* (PPI) of M30, using the STRING database (Szklarczyk et al., 2016). Using expression (or, as we called it in the paper, gene activity) data in the hippocampus from UKBEC data (Gene Expression Omnibus (GEO) accession number $GSE46706$), a Pearson's R correlation matrix is computed, which allows signing the interactions using pairwise correlation signs. Then, considering that the effects of a gene perturbation can only be seen on connected nodes, we only keep strongly connected components in which at least one gene perturbation in LINCS L1000 experiments occurs, using Tarjan's algorithm (Tarjan, 1972).

Then, in order to direct the edges in the network using the inference method, we restrict the experiments extracted from LINCS L1000 to those in SH-SY5Y human neuron cells (neuroblastoma from bone marrow), with a positive interference scale score (Cheng and Li, 2016), which quantify the success of the perturbation experiment. For each experiment (knockdown via shRNA, overexpression via cDNA) on a gene in M30 in this cell line, we extract from Level 3 LINCS same-plate untreated, genetic control and perturbed profiles (each of them being real-valued vectors of size ≈ 100, the number of genes in M30) such as the perturbed profile on this plate has the largest value of *distil_ss* which quantifies experimental replication. This procedure yields a total of (1 untreated + 2 replicates of genetic control + 2 replicates of perturbed) ×3 experimental profiles. Each experimental constraint for the GRN inference is defined as follows ($G = 101$ is the number of $M30$ genes in the network):

- **Initial condition**: Untreated profile which has been binarized using the `binarize_via_histogram` procedure (peak detection in histogram of gene expression values using persistent topology). Value: $\{0, 1, \perp\}^G$. The number of non-$\perp$ values is 66.
- **Perturbation**: Gene-associated value is equal to 1 if and only if the gene is perturbed in the experiment.
- **Final/fixpoint condition**: Vector which has been obtained by running Characteristic Direction (Clark et al., 2014) (CD) on [treated||genetic control] (in the call to the function, treated profiles were annotated 2 whereas genetic control ones were annotated 1) profiles, which yields a vector in $\{0, 1, \perp\}^G$, where 0's (resp. 1's) are significantly down- (resp. up-) regulated genes in treated profiles compared to control ones. Note that $[P_1||P_2]$ means that we compute the genewise activity change from group P_2 to group P_1 (hence, this is not symmetrical). The number of non-$\perp$ values is around 22.

The inferred GRN should satisfy all experimental constraints by assigning logical functions to genes and selecting gene interactions. This network is displayed in Figure 4.

Table 6: Experiments in SH-SY5Y human neuron-like cell line for GRN inference. Inference parameter: $t = 20$, asynchronous dynamics.

Perturbed gene	Experiment type	Exposure time
CACNA1C	KD	120 h
CDC42	KD	120 h
KCNA2	KD	120 h

Getting the arm/drug features The features we use are the drug signatures ($K = 509$ in the binary drug signature dataset). Given a drug, we compute them as follows:

1. First, in the Level 3 LINCS L1000 database (Subramanian et al., 2017), we select the cell line with the highest transcriptomic activity score, or TAS (quantifying the success of treatment in this specific cell line: we expect to obtain more reproducible experiments in this cell line if this score is high). Then, considering experiment in this cell line, treatment by the considered drug, we select the *brew_prefix* (identifier for experimental plate) which correspond to the treated expression profile with the largest value of *distil_ss* (which quantifies the reproducibility of the profile across replicates), and we get the corresponding same-plate control (vehicle) profile. We also get one same-plate replicate of the considered treated experiment and another of the control experiment (total number of profiles: $2 + 2$).
2. We apply on this set of profiles Characteristic Direction (Clark et al., 2014) in order to get the relative genewise expression change due to treatment from control sample group CD([treated||vehicle control]). This yields a real-valued vector in $[-1, 1]^G$ which will be used in the baseline method L1000 CDS[2], and a binary vector in $\{0, 1, \perp\}^G$ in our scoring method, which is the so-called drug signature.

Epileptic patient/control phenotypes We fetched data from GEO accession number $GSE77578$, which was then quantile-normalized across all patient ($|P_p| = 18$) and control ($|P_c| = 17$) samples. We run Characteristic Direction (Clark et al., 2014) $CD([P_c||P_p])$ in order to get the "differential phenotype" from controls to patients, which is the way we chose in order to aggregate control profiles and only considering differentially expressed genes.

Drug "true" scores We get them from RepoDB (Brown and Patel, 2017) database, which is a curated version from `clinicaltrials.gov`, and from literature. To each drug is associated an integer: 1 if the drug is antiepileptic, 0 if it is unknown, -1 if it is a proconvulsant drug.

Masking procedure $\rtimes$ We use this (asymmetric) function in order to generate the initial condition from which an attractor state, if it exists, should be fetched: $(x \rtimes y)[j] = y[j]$ if $y[j] \in \{0, 1\}$ else $x[j]$. This aims at mimicking the immediate effect of treatment on gene activity.

Running the simulator *via* the GRN Given collected patient and control phenotypes, and seeing arms as potentially repurposed drugs, the procedure to generate a reward from a given arm a is as described in Algorithm 2. We compare this method to a simpler signature reversion method, used in the web application L1000 CDS^2 (Duan et al., 2016), which is deterministic, and compares directly drug signatures and differential phenotypes. The full procedure is described in Algorithm 3. We have tested our method on a subset of drugs with respect to this baseline. The results can be seen in Figure 5.

Note that returning a score for a single drug with our method is usually a matter of a few minutes, but the computation time can drastically increase when considering a higher number of nodes in the Boolean network, so that is why, even if on this instance we could run all computations for each drug and for each initial patient sample for drug repurposing (which is what we do in Figure 5 anyway in order to check that our method yields correct results with respect to known therapeutic indications), we think this model is interesting to test our bandit algorithms.

Moreover, the linear dependency between features and scores does not hold: indeed, in our subset of $K = 10$ arms, computing the least squares estimate of θ using the mean rewards $\overline{m}$ as true values, and denoting X the concatenation of drug signatures, that is, $\theta = (X^\top X)^{-1} X\overline{m}$, gives a high value of $\|\theta - \overline{m}\| \approx 20.9$. The linear setting in bandits is simply the easiest contextual setting to analyze. Although this non-linearity, along with the fact that the initial condition is randomized, might be the main reason why the empirical sample complexity in our subset instance is a lot higher than 10×18 for all algorithms, even if linear algorithms are noticeably more performant than classical ones.

Algorithm 2 Reward generation via the "GRN simulator".

```
  requires $G$ GRN, phenotypes of patient (diseased) and control (healthy) individuals w.r.t. a given disease
  $P_p \in [0, 15]^G$, $P_c \in [0, 15]^G$, $a$ arm/drug to be tested, with binary drug signature $s_a^b \in \{0, 1, \perp\}^G$.
  # differential phenotype is computed: controls||patients
  $D \in \{0, 1, \perp\}^G = CD([P_c||P_p])$
  # patient phenotype is uniformly sampled from the pool of patient phenotypes
  $p \sim \mathcal{U}(P_p)$
  $p^b \in \{0, 1, \perp\}^G \leftarrow$ binarize_via_histogram($p$)
  $p^r \in \{0, 1\}^G \leftarrow$ phenotype_prediction(GRN=G, initial_condition=$(p^b \rtimes s_a^b) \in \{0, 1, \perp\}^G$)
  # comparison function cosine_score is run on the intersection of supports of $D$ and $p^r$
  # this intersection is equal to $50$ in practice, which is the size of the support of $D$
  $r \leftarrow$ cosine_score$_{|D|\cap|p^r|}(D, p^r)$
  returns $r$
```

Algorithm 3 Reward via baseline method from L1000 CDS^2 (Duan et al., 2016).

```
  requires Phenotypes of patient (diseased) and control (healthy) individuals w.r.t. a given disease $P_p \in [0, 15]^G$,
  $P_c \in [0, 15]^G$, $a$ arm/drug to be tested, with non-binary, full signature $s_a \in [-1, 1]^G$.
  # differential phenotype is computed: patients||controls
  $C \in \{0, 1, \perp\}^G = CD([P_p||P_c])$
  $r \leftarrow 1 -$ cosine_score$(C, s_a)$
  returns $r$
```

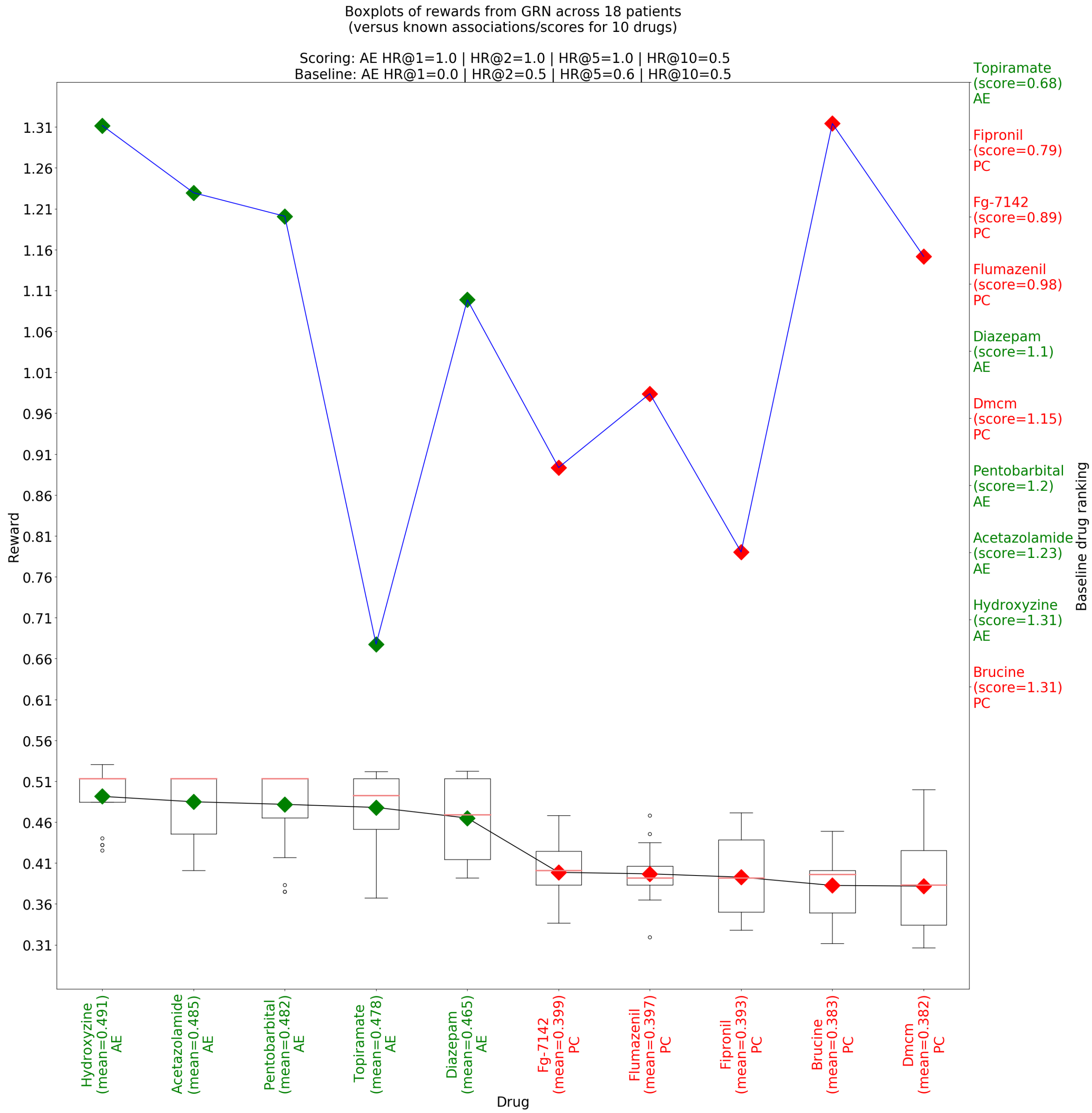

Figure 5: We consider a subset of drugs of size 10 (5 with positive association score, 5 with negative score), which is the one tested in the paper. For validation, we plot a boxplot of the rewards obtained for each initial patient sample, for each drug. Mean is colored as green if the drug is antiepileptic (AE), resp. red if it is proconvulsant (PC), with the corresponding drug name (in red if its true score is negative, in green if it is positive). The baseline score is plot in in blue. For both methods, the highest the score is, the better (the "more" anti-epileptic the drug is predicted). We computed and reported above the plot the Hit Score at rank r (HR@r), that is, the mean accuracy on the class AE on the Top-r scores, for $r \in \{1, 2, 5, 10\}$ for each of the methods (Scoring or Baseline).

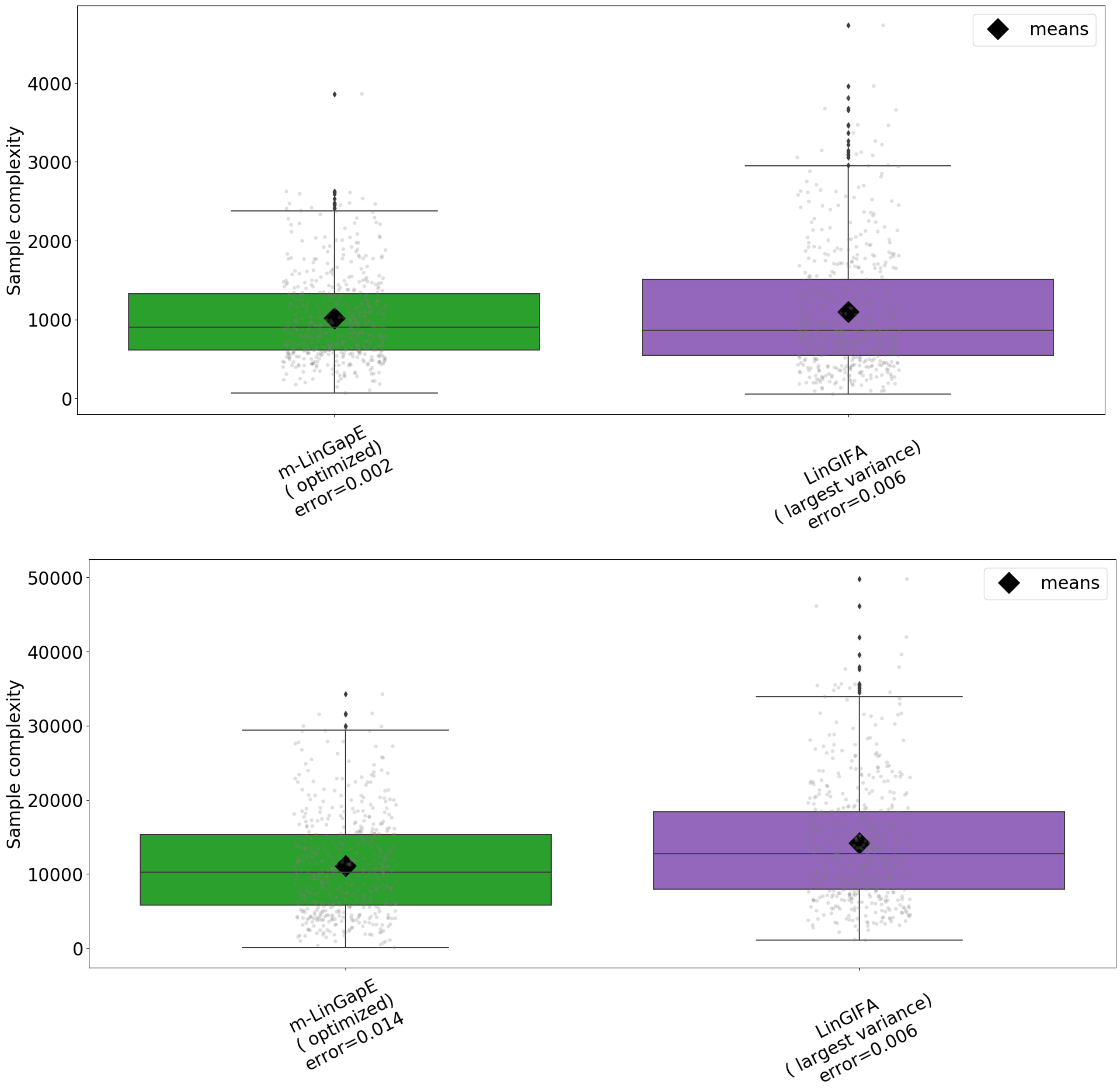

Figure 3: From top to bottom: $m = 3$, $m = 8$. Error frequencies are rounded up to 5 decimal places.

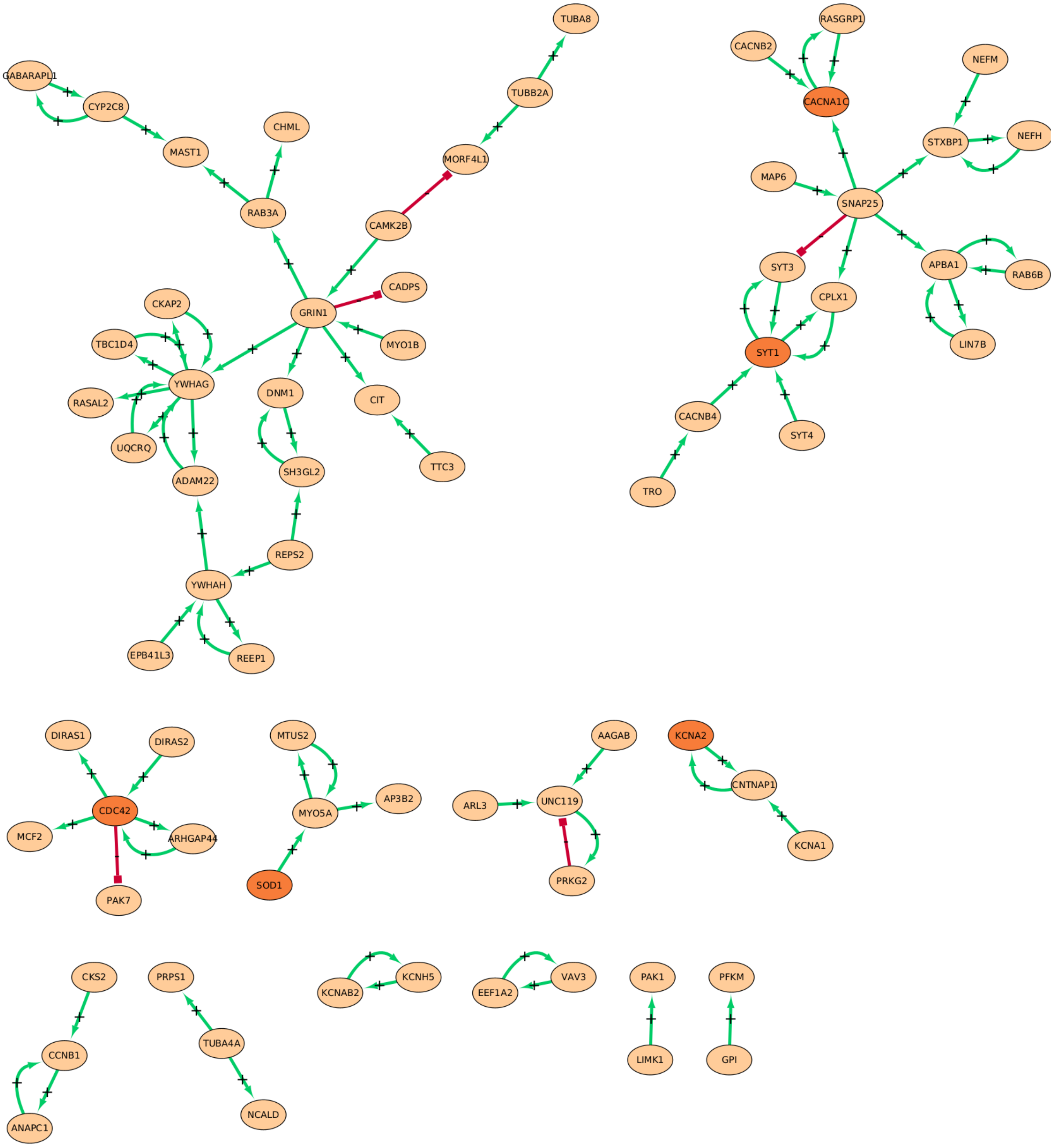

Figure 4: Inferred GRN for the drug repurposing instance on epilepsy. Genes present in this network belong to the M30 set. Green edges labelled "+" (resp. red edges labelled "-") are activatory (resp. inhibitory) interactions from one regulatory gene on a target gene, that is, that increase (resp. decrease) target gene activity. Deep orange nodes are perturbed nodes in the SHSY5Y cell line in LINCS L1000.

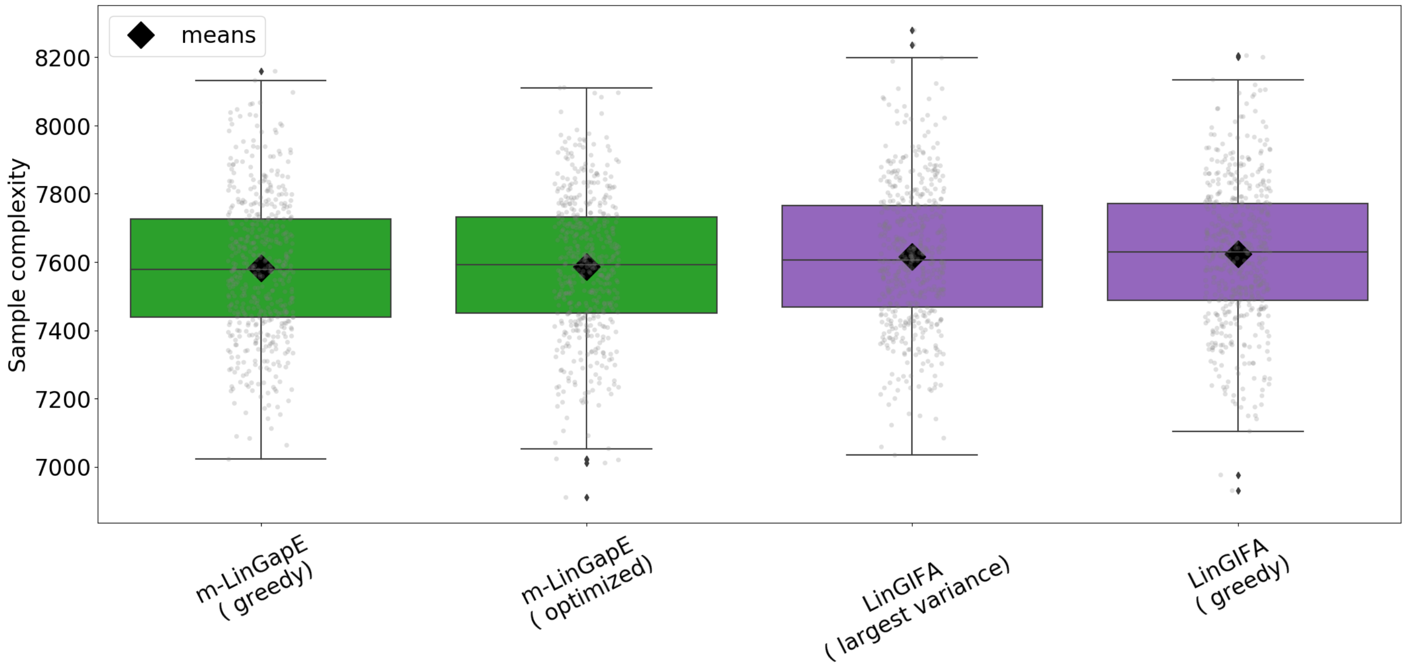

Figure 6: Drug repurposing instance $K = 10$, 500 simulations, $m = 5$, $\delta = 0.05, \epsilon = 0, \sigma = 0.5, \lambda = \sigma/20$. Close-up from Figure 1 from the paper.

C UPPER BOUNDS FOR GIFA ALGORITHMS

Lemma 7. *In algorithm m-LinGapE, for any selection rule, on event* $\mathcal{E} \triangleq \bigcap_{t>0} \bigcap_{i,j \in [K]} \Big(\Delta_{i,j} \in [-B_{j,i}(t), B_{i,j}(t)] \Big)$*, for all* $t > 0$*,* $B_{c_t,b_t}(t) \leq \min(-(\Delta_{b_t} \vee \Delta_{c_t}) + 2W_t(b_t,c_t), 0) + W_t(b_t,c_t)$*.* (Lemma 4 in the paper)

Proof. Let us use two properties:

1. As $b_t \in J(t)$ and $c_t \notin J(t)$, it holds in particular that $\hat{\mu}_{b_t}(t) \geq \hat{\mu}_{c_t}(t)$, hence $B_{c_t,b_t}(t) = \hat{\Delta}_{c_t,b_t}(t) + W_t(b_t,c_t) \leq W_t(b_t,c_t)$.

2. From the definitions of b_t and c_t, it holds that $B_{c_t,b_t}(t) = \max_{j \in J(t)} \max_{i \notin J(t)} B_{i,j}(t)$.

Property 1 already establishes that $B_{c_t,b_t}(t) \leq W_t(b_t,c_t)$, it therefore remains to show that $B_{c_t,b_t}(t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t)$. We do it by distinguishing four cases:

(i) $b_t \in \mathcal{S}^\star_m$ **and** $c_t \notin \mathcal{S}^\star_m$: In that case $\Delta_{b_t} = \mu_{b_t} - \mu_{m+1}$ and $\Delta_{c_t} = \mu_m - \mu_{c_t}$. As event $\mathcal{E}$ holds, one has $B_{c_t,b_t}(t) = -B_{b_t,c_t}(t) + 2W_t(b_t,c_t) \leq \Delta_{c_t,b_t} + 2W_t(b_t,c_t)$. As $c_t \notin \mathcal{S}^\star_m$, $\mu_{c_t} \leq \mu_{m+1}$, and $\Delta_{c_t,b_t} \leq \mu_{m+1} - \mu_{b_t} = -\Delta_{b_t}$. But as $b_t \in \mathcal{S}^\star_m$, it also holds that $\mu_{b_t} \geq \mu_m$, and $\Delta_{c_t,b_t} \leq \mu_{c_t} - \mu_m = -\Delta_{c_t}$. Hence $B_{c_t,b_t}(t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 2W_t(b_t,c_t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t)$.

(ii) $b_t \notin \mathcal{S}^\star_m$ **and** $c_t \in \mathcal{S}^\star_m$: Using Property 1:

$$B_{c_t,b_t}(t) \leq W_t(b_t,c_t) \leq \hat{\Delta}_{b_t,c_t}(t) + W_t(b_t,c_t) = B_{b_t,c_t}(t) = -B_{c_t,b_t}(t) + 2W_t(b_t,c_t) \leq \Delta_{b_t,c_t} + 2W_t(b_t,c_t)$$

as event $\mathcal{E}$ holds. One can show with the same arguments as in the previous case that $B_{c_t,b_t}(t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t)$.

(iii) $b_t \notin \mathcal{S}^\star_m$ **and** $c_t \notin \mathcal{S}^\star_m$: In that case, there must exist $b \in \mathcal{S}^\star_m$ that belongs to $J(t)^c$. From the definition of c_t, it follows that $B_{c_t,b_t}(t) \geq B_{b,b_t}(t)$. Hence, using furthermore Property 1, the definition of c_t, event $\mathcal{E}$ and $b \in \mathcal{S}^\star_m$, $W_t(b_t,c_t) \geq B_{c_t,b_t}(t) \geq B_{b,b_t}(t) \geq \Delta_{b,b_t} \geq \Delta_{m,b_t} = \Delta_{b_t}$. It follows that, using event $\mathcal{E}$:

$$\begin{aligned} B_{c_t,b_t}(t) &\leq \Delta_{c_t,b_t} + 2W_t(b_t,c_t) = (\mu_{c_t} - \mu_m) + (\mu_m - \mu_{b_t}) + 2W_t(b_t,c_t) = -\Delta_{c_t} + \Delta_{b_t} + 2W_t(b_t,c_t) \\ &\quad (b_t \notin \mathcal{S}^\star_m \text{ and } c_t \notin \mathcal{S}^\star_m) \\ &\leq -\Delta_{c_t} + 3W_t(b_t,c_t) \end{aligned}$$

And it also holds by Property 1 that:

$$\begin{aligned} B_{c_t,b_t}(t) &\leq W_t(b_t,c_t) = -W_t(b_t,c_t) + 2W_t(b_t,c_t) \\ &\leq -\Delta_{b_t} + 2W_t(b_t,c_t) \leq -\Delta_{b_t} + 3W_t(b_t,c_t) \end{aligned}$$

Hence $B_{c_t,b_t}(t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t)$.

(iv) $b_t \in \mathcal{S}^\star_m$ **and** $c_t \in \mathcal{S}^\star_m$: In that case, there must exist $c \notin \mathcal{S}^\star_m$ such that $c \in J(t)$. By Property 2, on event $\mathcal{E}$ and using $c \in (\mathcal{S}^\star_m)^c$, we know that $B_{c_t,b_t}(t) = \max_{j \in J(t)} \max_{i \notin J(t)} B_{i,j}(t) \geq \max_{i \notin J(t)} B_{i,c}(t) \geq B_{c_t,c}(t) \geq \mu_{c_t} - \mu_c \geq \mu_{c_t} - \mu_{m+1} = \Delta_{c_t}$. Hence, using furthermore Property 1 yields $\Delta_{c_t} \leq B_{c_t,b_t}(t) \leq W_t(b_t,c_t)$. It follows that, using event $\mathcal{E}$:

$$\begin{aligned} B_{c_t,b_t}(t) &\leq \mu_{c_t} - \mu_{b_t} + 2W_t(b_t,c_t) = \mu_{c_t} - \mu_{m+1} + \mu_{m+1} - \mu_{b_t} + 2W_t(b_t,c_t) \\ &= \Delta_{c_t} - \Delta_{b_t} + 2W_t(b_t,c_t) \leq -\Delta_{b_t} + 3W_t(b_t,c_t) \end{aligned}$$

And using again Property 1, one has:

$$\begin{aligned} B_{c_t,b_t}(t) &\leq W_t(b_t,c_t) = -W_t(b_t,c_t) + 2W_t(b_t,c_t) \leq -\Delta_{c_t} + 2W_t(b_t,c_t) \\ &\leq -\Delta_{c_t} + 3W_t(b_t,c_t) \end{aligned}$$

Hence $B_{c_t,b_t}(t) \leq -(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t)$, which is what we wanted to show. □

Lemma 8. Upper bound in m-LinGapE with either or both b_t and c_t pulled at time t (m-LinGapE(1)) *Maximum number of samplings on event $\mathcal{E}$ is upper-bound by* $\inf_{u\in\mathbb{R}^{*+}}\{u > 1 + H^{\varepsilon}(m\text{-}LinGapE(1),\mu)C^2_{\delta,u}\}$, *where* $H^{\varepsilon}(m\text{-}LinGapE(1),\mu) \triangleq 4\sigma^2 \sum_{a\in[K]} \max\left(\varepsilon, \frac{\varepsilon+\Delta_a}{3}\right)^{-2}$.

Proof. Combining Lemma 4 with stopping rule τ^{LUCB}, at time $t < \tau^{LUCB}$:

$$\begin{aligned} &\varepsilon \leq B_{c_t,b_t}(t) \leq \min(-(\Delta_{b_t} \vee \Delta_{c_t}) + 3W_t(b_t,c_t), W_t(b_t,c_t)) \\ \Leftrightarrow \max\left(\varepsilon, \frac{\varepsilon+\Delta_{b_t}}{3}, \frac{\varepsilon+\Delta_{c_t}}{3}\right) &\leq W_t(b_t,c_t) \leq W_t(b_t) + W_t(c_t) \leq 2W_t(a_t) = 2C_{\delta,t}||x_{a_t}||_{\hat{\Sigma}_t^{\lambda}} \\ &\text{(where } a_t = \max\nolimits_{a\in\{b_t,c_t\}} W_t(a)) \\ &= 2\sigma C_{\delta,t}||x_{a_t}||_{(\hat{V}_t^{\lambda})^{-1}} \leq 2\sigma C_{\delta,t}\frac{||x_{a_t}||}{\sqrt{N_{a_t}(t)||x_{a_t}||}} \\ &= 2\sigma C_{\delta,t}\frac{1}{\sqrt{N_{a_t}(t)}} \text{ (using Lemma 2 and } \lambda > 0, N_{a_t}(t) > 0\text{, since } a_t \text{ is pulled at } t) \\ \Leftrightarrow N_{a_t}(t) &\leq \frac{4\sigma^2 C^2_{\delta,t}}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_{b_t}}{3}, \frac{\varepsilon+\Delta_{c_t}}{3}\right)^2} \leq \min_{a\in\{b_t,c_t\}} \frac{4\sigma^2 C^2_{\delta,t}}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_a}{3}\right)^2} \leq \frac{4\sigma^2 C^2_{\delta,t}}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_{a_t}}{3}\right)^2} \\ \Leftrightarrow N_{a_t}(t) &\leq \frac{4\sigma^2 C^2_{\delta,t}}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_{a_t}}{3}\right)^2} = T^*(a_t,\delta,t) \end{aligned}$$

Using Lemma 6, if $T(\mu,\delta)$ is the number of samplings of m-LinGapE on bandit instance μ for δ-fixed confidence Top-m identification:

$$T(\mu,\delta) \leq \inf_{u\in\mathbb{R}^{*+}}\left\{u > 1 + C^2_{\delta,u}\sum_{a\in[K]}\frac{4\sigma^2}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_a}{3}\right)^2}\right\} \leq \inf_{u\in\mathbb{R}^{*+}}\{u > 1 + C^2_{\delta,u}\mathrm{H}^{\varepsilon}(m\text{-LinGapE}(1),\mu)\}$$

□

D TECHNICAL LEMMAS

Lemma 9. *Let us fix $K > m > 0$, $t > 0$ and $i \in [K]$. Let us consider μ such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$, and a series of distinct values $(B_{j,i}(t))_{j\in[K]}$ such that $B_{j,i}(t) \geq \mu_j - \mu_i$ for any $j \in [K]$. Then $\overset{m}{\max_{j\in[K]}} B_{j,i}(t) \geq \mu_m - \mu_i$.*

Proof. Assume by appealing to the extremes that $\overset{m}{\max_{j\in[K]}} B_{j,i}(t) < \mu_m - \mu_i$. Then, using our assumption on $(B_{j,i}(t))_{j\in[K]}$ and $(\mu_j)_{j\in[K]}$, for any $j \in [m]$, $B_{j,i}(t) \geq \mu_j - \mu_i \geq \mu_m - \mu_i > \overset{m}{\max_{j\in[K]}} B_{j,i}(t)$, which means at least m distinct values of $(B_{j,i}(t))_{j\leq K}$ are strictly greater than $\overset{m}{\max_{j\in[K]}} B_{j,i}(t)$, which yields a contradiction. Thus $\overset{m}{\max_{j\in[K]}} B_{j,i}(t) \geq \mu_m - \mu_i$. Note that we can assume the condition on $(B_{i,j}(t))_{i,j\in[K]}$ being distinct is satisfied except for some degenerate cases where two arm features are equal and the observations made from both arms are exactly the same. □

Lemma 10. *For all $t > 0$, for any subset $J \subseteq [K]$ of size m, for all $j \in J$, $\overset{m}{\max_{i \neq j}} B_{i,j}(t) \leq \max_{i \notin J} B_{i,j}(t)$.* (Lemma 1 in the paper)

Proof. Indeed, $\overset{m}{\max_{i \neq j}} B_{i,j}(t) = \min_{S \subseteq [K], |S|=m-1} \max_{i \notin (S \cup \{j\})} B_{i,j}(t)$ (set S matching the outer bound is $\arg \overset{[m-1]}{\max_{i \neq j}} B_{i,j}(t)$, meaning that we consider then the maximum value over the set of $(B_{i,j}(t))_{i \in [K]}$ from which the $m-1$ largest values and $B_{j,j}(t)$ are removed). Then consider $S = J \setminus \{j\}$, which is included in $[K]$ and is of size $m-1$ ($j \in J$). Then $\overset{m}{\max_{i \neq j}} B_{i,j}(t) \leq \max_{i \notin S \cup \{j\}} B_{i,j}(t) = \max_{i \notin J} B_{i,j}(t)$. □

Lemma 11. *For any $t > 0$, for any $a \in [K]$ such that $N_a(t) > 0$, for all $x \in \mathbb{R}^N$, $||x||^2_{(\hat{V}_t^\lambda)^{-1}} \leq x^\top(\lambda I_N + N_a(t)x_a x_a^\top)^{-1}x$.*

Proof. Let us prove this lemma by induction on $K \geq 2$ (case $K = 1$ is trivial). Let $A_t(a) \triangleq N_a(t)x_a x_a^\top$, $A_t \triangleq \lambda I_N + A_t(a)$. For $[K] = \{a_1, a_2, \ldots, a_{K-1}, a\}$ and $K \geq 2$, let us denote $B_K^t \triangleq \sum_{i=1}^{K-1} A_t(a_i)$, such that $\hat{V}_t^\lambda = A_t + B_K^t$. We will prove a stronger claim, which is "for any $t \in \mathbb{N}^*$, for any $x \in \mathbb{R}^N$, and $K \geq 2$, A_t and $A_t + B_K^t$ are invertible and $||x||^2_{(A_t+B_K^t)^{-1}} < ||x||^2_{A_t^{-1}}$". Note that, for any K and t, since $\lambda > 0$, A_t is then a Gram matrix with linearly independent columns, thus is positive definite, and B_K^t is a Gram matrix, thus a non-negative definite matrix. Then $A_t + B_K^t$ and A_t are positive definite and invertible.

<u>If $K = 2$:</u> then let us assume that $[K] = \{a, a_1\}$:

$$\begin{aligned}||x||^2_{(A_t+B_2^t)^{-1}} &\triangleq x^\top(A_t + B_2^t)^{-1}x = x^\top(A_t + N_{a_1}(t)x_{a_1}x_{a_1}^\top)^{-1}x \text{ (using Sherman-Morrison formula)}\\ &= x^\top(A_t^{-1} - \frac{A_t^{-1}N_{a_1}(t)x_{a_1}x_{a_1}^\top A_t^{-1}}{1 + N_{a_1}(t)||x_{a_1}||^2_{A_t^{-1}}})x = ||x||^2_{A_t^{-1}} - \frac{(A_t^{-1}x)^\top B_2^t(A_t^{-1}x)}{1 + N_{a_1}(t)||x_{a_1}||^2_{A_t^{-1}}} \leq ||x||^2_{A_t^{-1}} - 0\end{aligned}$$

using the fact that B_2^t is nonnegative definite and A_t, and then A_t^{-1}, are both symmetric.

<u>If $K > 2$:</u> using the induction, $A_t + B_{K-1}^t$ is invertible. Similarly to the previous step, using the Sherman-Morrison formula:

$$\begin{aligned}||x||^2_{(A_t+B_K^t)^{-1}} &= x^\top(A_t + B_{K-1}^t)^{-1}x - \frac{x^\top(A_t + B_{K-1}^t)^{-1}A_t(a_{K-1})(A_t + B_{K-1}^t)^{-1}x}{1 + N_{a_{K-1}}(t)||x_{a_{K-1}}||^2_{(A_t+B_{K-1}^t)^{-1}}}\\ &= x^\top(A_t + B_{K-1}^t)^{-1}x - \frac{((A_t + B_{K-1}^t)^{-1}x)^\top A_t(a_{K-1})((A_t + B_{K-1}^t)^{-1}x)}{1 + N_{a_{K-1}}(t)||x_{a_{K-1}}||^2_{(A_t+B_{K-1}^t)^{-1}}}\\ &\text{(same argument as previously, since } A_t(a_{K-1}) \text{ is a Gram matrix)}\\ &\leq x^\top(A_t + B_{K-1}^t)^{-1}x - 0 = ||x||^2_{(A_t+B_{K-1}^t)^{-1}}\end{aligned}$$

Then, using the induction, $||x||^2_{(\hat{V}_t^\lambda)^{-1}} = ||x||^2_{(A_t+B_K^t)^{-1}} \leq ||x||^2_{(A_t+B_{K-1}^t)^{-1}} \leq ||x||^2_{A_t^{-1}} = ||x||^2_{(\lambda I_N + N_a(t)x_a x_a^T)^{-1}}$.

□

Lemma 12. $\forall t > 0, \forall a \in [K], \forall y \in \mathbb{R}^N$, $||y||_{(\hat{V}_t^\lambda)^{-1}} \leq ||y||/\sqrt{N_a(t)||x_a||^2 + \lambda}$. (Lemma 2 in the paper)

Proof. Using successively Lemma 11, Sherman-Morrison formula and Cauchy-Schwarz inequality:

$$\begin{aligned}||y||^2_{(\hat{V}_t^\lambda)^{-1}} \leq ||y||^2_{(\lambda I_N + N_a(t)x_a x_a^\top)^{-1}} &= \frac{||y||^2}{\lambda} - \frac{\lambda^{-2}N_a(t)(< y, x_a >)^2}{1 + \lambda^{-1}N_a(t)||x_a||^2}\\ &\leq \frac{||y||^2}{\lambda} - \frac{\lambda^{-2}N_a(t)||y||^2||x_a||^2}{1 + \lambda^{-1}N_a(t)||x_a||^2} = \frac{||y||^2}{\lambda + N_a(t)||x_a||^2}\end{aligned}$$

□

Lemma 13. *Let $T^* : [K] \times (0,1) \times \mathbb{N}^* \rightarrow \mathbb{R}^{*+}$ be a function that is nondecreasing in t, and $\mathcal{I}_t$ the set of pulled arms at time t. Let $\mathcal{E}$ be an event such that for all $t < \tau_\delta, \delta \in (0,1)$, $\exists a_t \in \mathcal{I}_t, N_{a_t}(t) \leq T^*(a_t, \delta, t)$. Then it holds on the event $\mathcal{E}$ that $\tau_\delta \leq T(\mu, \delta)$ where*

$$T(\mu,\delta) \triangleq \inf\left\{u \in \mathbb{R}^{*+} : u > 1 + \sum_{a=1}^{K} T^*(a,\delta,u)\right\}.$$

(Lemma 6 in the paper)

Proof. Let us denote $T \in \mathbb{N}^*$. Let us study $\min(\tau_\delta, T)$, because $\min(\tau_\delta, T) < T \implies \tau_\delta < T$. On event $\mathcal{E}$:

$$\begin{aligned}
\min(\tau_\delta, T) &= 1 + \sum_{t \leq T} \mathbb{1}(t < \tau_\delta) \leq 1 + \sum_{t \leq T} \mathbb{1}(\exists a_t \in \mathcal{I}_t, N_{a_t}(t) \leq T^*(a_t,\delta,t)) \text{ (using definition of } T^*\text{, and } \mathcal{E} \text{ holds)} \\
&= 1 + \sum_{t \leq T} \sum_{m=1}^{t} \mathbb{1}(\exists a_t \in \mathcal{I}_t, N_{a_t}(t) = m \wedge m \leq T^*(a_t,\delta,t)) \text{ (using } \forall a \in [K], N_a(t) \in [t] \wedge \forall a \in \mathcal{I}_t, N_a(t) > 0) \\
&\leq 1 + \sum_{m=1}^{T} \sum_{t=m}^{T} \sum_{a \in [K]} \mathbb{1}(a \in \mathcal{I}_t)\mathbb{1}(N_a(t) = m \wedge m \leq T^*(a,\delta,t)) \text{ (using the union bound on pulled arms)} \\
&= 1 + \sum_{a \in [K]} \sum_{m=1}^{T} \sum_{t=m}^{T} \mathbb{1}(a \in \mathcal{I}_t)\mathbb{1}(N_a(t) = m)\mathbb{1}(m \leq T^*(a,\delta,t)) \\
&\leq 1 + \sum_{a \in [K]} \sum_{m=1}^{T} \left[\sum_{t=m}^{T} \mathbb{1}(a \in \mathcal{I}_t)\mathbb{1}(N_a(t) = m)\right] \mathbb{1}(m \leq T^*(a,\delta,T)) \text{ (since } T^* \text{ is nondecreasing in } t) \\
&\leq 1 + \sum_{a \in [K]} \sum_{m=1}^{T} 1 \times \mathbb{1}(m \leq T^*(a,\delta,T)) \leq 1 + \sum_{a \in [K]} T^*(a,\delta,T)
\end{aligned}$$

Choosing any T that satisfies $1 + \sum_{a \in [K]} T^*(a,\delta,T) < T$ yields $\min(\tau_\delta, T) < T$ and therefore $\tau_\delta \leq T$. The smallest possible such T is

$$T(\mu,\delta) \triangleq \inf\left\{u \in \mathbb{R}^{*+} : u > 1 + \sum_{a \in [K]} T^*(a,\delta,u)\right\}.$$

□