This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

Q-Rank: Reinforcement Learning for Recommending Algorithms to Predict Drug Sensitivity to Cancer Therapy

Salma Daoud, Afef Mdhaffar, Mohamed Jmaiel, and Bernd Freisleben

***Abstract*—In personalized medicine, a challenging task is to identify the most effective treatment for a patient. In oncology, several computational models have been developed to predict the response of drugs to therapy. However, the performance of these models depends on multiple factors. This paper presents a new approach, called Q-Rank, to predict the sensitivity of cell lines to anti-cancer drugs. Q-Rank integrates different prediction algorithms and identifies a suitable algorithm for a given application. Q-Rank is based on reinforcement learning methods to rank prediction algorithms on the basis of relevant features (e.g., omics characterization). The best-ranked algorithm is recommended and used to predict the response of drugs to therapy. Our experimental results indicate that Q-Rank outperforms the integrated models in predicting the sensitivity of cell lines to different drugs.**

***Index Terms*—prediction, reinforcement learning, algorithm recommendation, precision medicine, drug response to therapy**

I. Introduction

THE accuracy of prediction models for predicting the response of drugs to therapy based on the molecular profiles of patients is a crucial factor in precision medicine. Prediction models help clinicians to select the most effective therapeutic choice available and provide recommendations to select suitable patients in clinical trials [1].

Several prediction algorithms have been developed in oncology. They are based on the use of various computational approaches and heterogeneous omics data types. These tools improve quality of life and reduce costs [2].

However, several studies [3], [4], [5] have shown that the performance of prediction algorithms depends on many factors. To select a suitable prediction algorithm for a given application, we need to identify key factors impacting the performance of these algorithms and then use the identified factors to recommend a prediction algorithm for the given application. Several parameters influencing the performance of prediction algorithm in cancer cell line sensitivity were identified in previous studies [1], [3], [4], [5], [6], [7]. However, domain expertise, excessive training, test, and cross-validation are needed to identify the most suitable weights of each parameter and its impact on algorithm performance.

S. Daoud, A. Mdhaffar and M. Jmaiel are with the Department of Computer Engineering and Applied Mathematics, ENIS, University of Sfax, Tunisia. E-Mail: {salma.daoud, afef.mdhaffar, mohamed.jmaiel}@redcad.org

B. Freisleben is with the Department of Mathematics and Computer Science, University of Marburg, Germany. E-Mail: freisleben@uni-marburg.de

Manuscript received November 15, 2019

In our previous work [6], we used benchmark studies to identify the weight of each feature and provided interactive decision making methods to recommend a suitable algorithm for each application.

In this paper, a fully automated prediction approach called Q-Rank is presented. Q-Rank predicts drug sensitivity using existing prediction models. Hence, for each case, the integrated models are automatically ranked on the basis of non-scored meta-features. The ranking policy is identified using batch reinforcement learning. The top-ranked model(s) is (are) used to predict drug responses. The ranking policy is improved using online reinforcement learning (i.e., policy iteration), whenever new data is provided (i.e., the health professionals align the database by new clinical results of cancer therapies).

Experimental results demonstrate the merits of Q-Rank in terms of accuracy, precision, recall, and F1-measure, when compared to MCRM4Pred [6] and 3 other algorithms [8], [9], [10]. Indeed, Q-Rank improves the accuracy by 5% compared to KBMTL and MCRM4Pred, and 8% compared to IntegratedMRF, on the average. Through k-fold cross validation, Q-Rank reached accurate results for predicting drug sensitivity to cancer therapy. The contributions of our work are:

1) We formulate the ranking of machine learning algorithms as a reinforcement learning problem.
2) We determine the suitability of machine learning algorithms for predicting sensitivity to cancer therapy in a fully automated manner.
3) We present a set of experiments that show the impact of each meta-feature on algorithm performance. This classification allows clinicians to identify a suitable algorithm for future applications.
4) We improve the accuracy of drug sensitivity predictions, which is the main objective of precision medicine.

The paper is organized as follows. Section II introduces the background of this work. In Section III, an overview of related work is presented. Section IV presents Q-Rank, with a mathematical formulation and a detailed description of its architecture. Implementation details are presented in Section V. Our experimental results are discussed in Section VI. Finally, Section VIII concludes the paper and outlines areas for future research.

II. Background

In this section, basic concepts of reinforcement learning (RL) methods are briefly reviewed. Then, drug sensitivity measurements in cancer cell lines are explained.

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

A. Reinforcement Learning

Machine learning (ML) allows systems to automatically learn and improve from experience without being explicitly programmed. Reinforcement Learning is a field of machine learning to solve sequential and stochastic decision making problems [11]. In RL, the goal of the agent is to retrieve a policy by maximizing the expected sum of discounted rewards. For this purpose, RL makes use of an agent that sequentially selects different actions based on the states. For each action, it receives a feedback in the form of a numerical reward that needs to be maximized over time. Therefore, the agent applies the action that returns the highest long-term reward. Occasionally, it follows a random choice as part of its exploration.

The mathematical model of reinforcement learning is a Markov decision process (MDP). An MDP is a tuple $M = (S, A, R, T, \gamma)$, where S is the state space, A is the action space of the agent, R is the reward function, T is the state transition function and $\gamma \in [0, 1]$ is the discount rate. The objective of an agent in an MDP is to find an optimal policy that maximizes the expected accumulative rewards starting from any state s that is defined by the optimal $state - value$ function: $V^*(s) = max_\pi \mathbb{E}^\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s\}$

where:

$\pi : S \times A \Rightarrow [0, 1]$ denotes any policy of the agent,

$\mathbb{E}^\pi$ is the expectation under policy π,

t is the current time step, k is a future time step,

and r_{t+k} is the immediate reward at the time step $(t + k)$.

This goal is equivalent to finding the optimal $action-value$ $Q^*(s, a) = max_\pi \mathbb{E}^\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a\}$, for any state-action pair (s, a). Hence, the optimal policy can be found by solving the Bellman equation:

$Q^*(s, a) = R(s, a) + \gamma max_{a'} \sum_{s'} T(s^{'}|a, s) Q^*(s^{'}, a^{'})$

RL techniques can be classified into two groups: online and batch (i.e., offline) learning. In batch mode, a learning algorithm is trained using the entire data set to generate a predictor. The predictor is deployed and applied during the entire runtime of the application (i.e., without performing any "updates" during its execution).

Online learning is a subfield of ML intended to learn models incrementally from data in a sequential manner. The goal is to ensure that the online learner makes an accurate decision by using the knowledge (i.e., correct answers) from previous prediction.

Several methods, e.g., Temporal Difference [11], compute the $Q^*(s, a)$ values by directly by interacting with the environment in a trial-and-error manner. For each step presented by the tuple $(st, at, rt, st+1)$, the $action-values$ are estimated by an online algorithm, such as Q-learning [11].

Although model-free online learning methods have been very successful when applied to problems with small discrete state spaces, they may cause slow learning in practice when applied to systems with larger state spaces. Several studies have focused on developing batch RL algorithms [12]. The batch mode allows us to effectively exploit the information included in the collected samples. This means that using small data sets is sufficient to obtain excellent performance [13].

B. Response to Treatment

Predicting the response of a specific cancer to a given therapy is a major challenge. The preclinical prediction method is a biological experiment used to predict the efficiency of a drug to hundreds of cells in parallel. Usually, after three days of treatment, the reaction of the treated cells to the drug can be read (i.e., the number of active cells for each drug concentration) [7]. If the drug is efficient as the drug's concentration increases, the number of cells decreases. Thus, in order to assess the response of these cell lines to treatment, multiple measurements may be extracted. Some of the metrics that are often used are (1) Effective Concentration EC50, which is the midpoint between no effect and the maximum effect recorded, (2) Inhibitory Concentration IC50, which is the midpoint between no effect and the absolute effect of killing all cells, and (3) Growth Inhibition GI50, which is the concentration for 50% of maximal inhibition of cell proliferation. On the basis of pre-clinical prediction results and the patient's omics characterization for each cell, we can better understand the dependency between genetics and its sensitivity to drug responses. Thus, the cell line sensitivity or resistance to drug treatment can be computationally predicted.

III. Related Work

A variety of approaches have been used to develop computational models for drug response prediction, such as linear regression models [14], neural network models [15], support vector machines [16], [17], matrix factorization methods [8], [18], [10], random forests [9], [19] and collaborative filtering methods [20], [21].

The main goal of the developed models in precision medicine is to return the highest accurate drug response prediction. Several studies [3], [4], [5], [6], [22] assessed these approaches and recommended the best algorithms for future applications.

Jang et al. [3] analyzed 110.000 models based on different factors, such as the type of molecular features, the compound and the methodology of predictions. These models are divided into two groups: (1) regression models that predict numerical values of drug sensitivity, and (2) classification models that classify cell lines as sensitive or resistant, according to drug sensitivity levels. Although regression models offer good performance in different settings, it has been demonstrated that their performance depends on the input data as well as the used measure of drug sensitivity (e.g., IC50, activity areas). Thus, there is no single approach that consistently outperforms all others in different settings.

Furthermore, a study [4] known as "NCI-DREAM challenge reviewed the computational models with the purpose of identifying top performing methods for predicting therapeutic response from omics data. They divided the 44 algorithms into six categories: (i) kernel methods, (ii) non-linear regression, (iii) sparse linear regression, (iv) partial least squares regression, (v) model selection and (vi) other that do not fall to any category. In this study [4], the authors found that none of the categories consistently outperformed the others over all experiments. This proves that the model's performance

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

is highly related to other factors, such as input data and algorithm's specific implementations.

Furthermore, Chen et al. [5] evaluated 17 prediction algorithms for the last 5 years on 4 datasets based on 9 metrics of performance. The authors evaluated the prediction methods, their interpretability, the importance of the integration of multiple omic profiles, the drug structure and pathways for performance improvement. After assessment, they divided the methods into three groups: good, unstable, and poor methods. They showed that good methods mostly outperformed the others and poor methods generally had the lowest performance. However, unstable methods presented inconclusive (i.e., changeable) efficiency, depending on the integrated omic profiles, drug, and the performance metrics. Indeed, the evaluation of these methods depend on (1) performance measurements and (2) supported data (e.g., omic characterization integrated in the model, drug structure/pathways).

Although these studies [3], [4], [5], [22] have assessed a comprehensive benchmarked set of algorithms for predicting therapeutic response based on multiple factors and provided a guide for future application and research, it has been shown that there is no generic solution that fits for all cases. Thus, a challenging task consists of combining a variety of existing prediction models and identifying the top-performing model for each case.

For this purpose, Daoud et al. [6] used comparative studies to identify the meta-feature impacting the performance of algorithms and provided interactive decision making methods to recommend a suitable algorithm for each application. However, in this study the weights of the identified meta-features were computed based on previous experimental results. Furthermore, their evaluation was mainly based on the drug type.

In this paper, ML is introduced to automatize weight attribution. To the best of our knowledge, no other work provides a fully automated recommendation system that applies several non-scored features to recommend a prediction algorithm for drug sensitivity.

IV. Q-RANK

This section presents a novel approach, called Q-Rank, that uses RL techniques to rank prediction algorithms, using a set of given meta-features for a given case. Then, the prediction of drug responses are computed based on a configurable recommendation method (e.g., using the top-ranked algorithm). Since RL approaches are formulated as an MDP problem, the Q-Rank MDP specifications are presented in Section IV-A. Q-Rank runs in two modes: (1) exploration mode, and (2) prediction mode. The exploration mode is first launched for data preparation and policy search. The prediction mode is based on the results of the exploration mode (i.e., identified policy) to generate a ranked list of candidate prediction algorithms. The exploration mode is re-launched when new data is collected. These modes are detailed in Section IV-B.

A. Problem Specification

Q-Rank is a learning to rank approach for prediction algorithms where the ranking process is formulated as a Markov decision process (MDP). This process is presented as a sequential decision making problem where each time step corresponds to a ranking position and each action is placing an algorithm for the corresponding position. Q-Rank is defined by the tuple (S, A, T, R, π) composed by States (S), Actions (A), Transition (T), Reward (R), and Policy (π).

Based on the state s_t in step t, the agent chooses an action a_t from the set of available actions for the given state s_t. Afterwards, the agent receives a numerical reward r_t related to its decision. Then, it moves to the next state s_{t+1}. The objective of RL is to identify a policy $\Pi(s_t, a_t)$ that determines which action a_t to take in state s_t. Figure 1 presents the MDP graph of Q-Rank. Here, two meta-features based on assessment results of several prediction models [2], [3], [4], [5], [6], [7] are considered, and three algorithms are integrated.

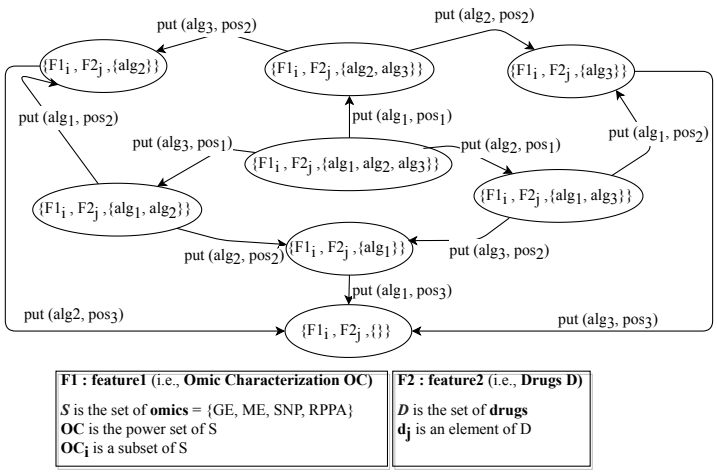

Fig. 1. Representation of Q-Rank as a Markov decision process

States $S = \{s_1, s_2 ... s_n\}$ is the set of states that describes the prediction query. At time step t, the state s_t is defined as a combination of meta-features and computational models: $s_t = [F1_i, F2_j, X_t]$ where:

- $F1_i$ is the first meta-feature, i.e., the type(s) of omic characterization C_i of the cell line
- $F2_j$ is the second meta-feature, i.e., the drug D_j
- X_t is the set of candidate algorithms at step t.

Actions A is a set of all actions that could be selected by an agent. An action a_t means put an algorithm x_i in position t.

Transition T (s,a) is a function $T : s \times a \mapsto s_{t+1}$ which maps a state s_t into a new state s_{t+1} in response to the selected action a_t. At the time step t, selecting an action a_t means: (1) associating the algorithm x_t to the ranking position t ; (2) removing the algorithm x_t from the candidates set (i.e., X_t), as shown in Equation (1):

$$s_{t+1} = [C_i, D_j, X_t \backslash x_t] \tag{1}$$

Reward $R(S, A)$ is the immediate reward or reinforcement. It evaluates the action of selecting an algorithm in the given position. The reward function uses the [pre]clinical[1] results to

[1][pre]clinical means pre-clinical or clinical

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

evaluate the accuracy of the prediction made by the algorithm. This function returns -1, if the selected algorithm provided a false prediction (FP, FN or non-conclusive), +1 if the selected algorithm provided a true prediction (TP or TN), otherwise, it returns 0 (i.e., cases that have missing values in the pre-clinical responses). Afterwards, the long-term reward is calculated. It is based on the normalized discounted cumulative gain value (NDCG), which allows us to find a good ranking of the algorithm by giving a higher weight by positions. Thus, the main goal is to determine a policy that maximizes the cumulative reward of an episode.

Policy $\pi(s_t, a_t)$ is a probabilistic distribution of the possible actions. $\pi(s_t, a_t)$ presents the behaviors of the agent. The policy defines which action a_t to take in state s_t. The goal of the agent is to learn the optimal policy function for all states and actions. A common approach to find the optimal policy is Q-learning [23]. Q-learning is based on the results of past experiences to continuously update the action values $Q(s_t, a_t)$. The policy $\pi = \arg\max Q(s_t, a_t)$ where $Q(s_t, a_t)$ is the state-action function specifying the expected reward for each possible action a_t in state s_t.

B. Q-Rank Modes

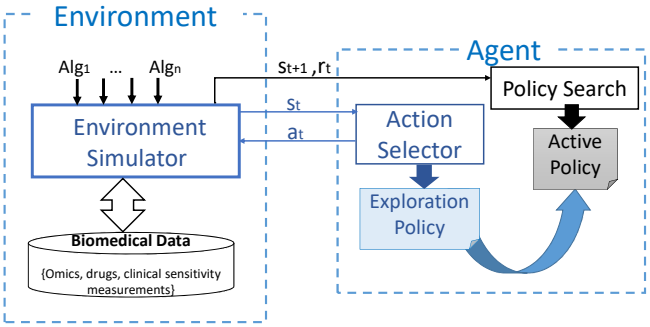

Fig. 2. Q-Rank: agent-environment interaction

The agent environment interaction for ranking algorithms is described in Figure2. Based on the state s_t, the agent chooses an action a_t using the exploration policy. The action a_t stands for attributing the algorithm (i), the position (t), where $i \in [1..P]$; P is the number of algorithms.

Afterwards, the environment computes a numerical reward r_t as an evaluation of the chosen algorithm. Then, it moves to the next state s_{t+1}. The agent uses the given reward to re-evaluate its policy called active policy. The active policy is used in the application (prediction mode).

In the following, the functioning modes of Q-Rank are explained, namely, the exploration mode and the prediction mode.

1) Exploration Mode: The main goal for the exploration is to identify the optimal policy using Batch RL. For this purpose, this mode collects knowledge about the value of candidate algorithms in each state using [pre]clinical data results. Collected data is used to learn the optimal policy. Figure 3 shows the exploration mode. Its life-cycle follows two phases.

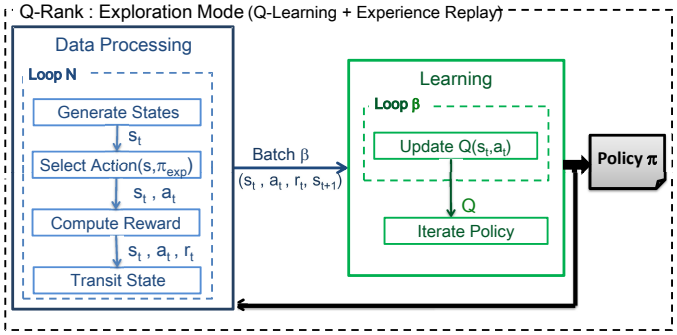

Fig. 3. Exploration mode: data processing and policy identification

i. Phase 1: Data Processing

First, an arbitrary sampling strategy is used to generate states. Then, a predefined exploration policy π_{exp} is applied to place a prediction algorithm in the right position for each state. Next, a feedback reward r_t is deduced on the basis of the comparison between obtained results regarding the sensitivity of the drugs using the selected predictive algorithm and [pre]clinical data results. Using the transition function T, the next state s_{t+1} is identified. The transition (s_t, a_t, r_t, s_{t+1}) is saved in a replay buffer called $\beta = \{tr : tr = (s_t, a_t, r_t, s_{t+1})\}$ (see Algorithm 1).

Algorithm 1: Data processing

require: states S, action A, reward function R, transition function T, active policy π
input : Batch size N
output : Batch β of (s, a, r, s')

begin
  for $i \leftarrow 1$ **to** N **do**
    Take a state s_t from the set S
    Select an action a_t from A using π

$$\pi = \begin{cases} Random, & \text{if } policy\ is\ not\ derived \\ \epsilon - Greedy, & otherwise \end{cases}$$

    Transit to the next state $T : s \times a \mapsto s_{t+1}$
    $s_{t+1} \leftarrow \{C, D, X_t \setminus alg_i\}$
    Compute the feedback $r_t = R(s_t, a_t)$
    Add the mini-batch (s_t, a_t, r_t, s_{t+1}) to β
  end for
  return β
end

i. Phase 2: Learning

In the learning phase, the replay buffer β is used to learn a state-action function (i.e., Q-learning). The result of the learning step is the best possible policy for every state-transition in the input data. To speed up the convergence of Q, the replay buffer β is repeatedly reused by the agent, as if they were new observations collected while interacting with a system. This phenomenon, known as *experience replay*, can speed up the convergence of the Q function by applying

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

the back propagation of feedback from current states to preceding states. Q is considered to converge towards the optimal solution when the learning curve gets flat and no longer increases. Therefore, Q-iteration is tracked. Once it becomes smaller than a predefined threshold, we stop executing iterations. Algorithm 2 details the learning process.

Algorithm 2: Q-Learning using a replay buffer

require: learning rate α ,discount factor γ, learning rule
input : Batch β
output : The agent policy π
begin
  Initialize $Q : X \times A \Rightarrow 0$
  while *Q is not converged* **do**
    Sample a transition from β uniformly
    Update Q values $Q(s_t, a_t) = Q(s_t, a_t) + \alpha \times (r_t, +\gamma \times \max a(Q(s_{t+1}, a)) - Q(s_t, a_t))$
  end while
end

2) Prediction Mode: This mode makes use of the active policy π that is identified during the exploration mode to rank algorithms. If the exploration mode has not been executed, the active policy could be initialized to an algorithm chosen by the developer or set to random. It should be pointed out that the active policy is dynamically updated. For this purpose, whenever new [pre]clinical data are available, the exploration mode is triggered allowing the system to update (i.e., iterate) the active policy.

The prediction mode follows two phases, as shown in Figure 4. The first phase allows us to rank the prediction algorithms using the active policy for a specific case. In the second phase, a recommendation method is applied (e.g., top ranked algorithm is recommended and used) to predict the sensitivity of cell lines to drugs.

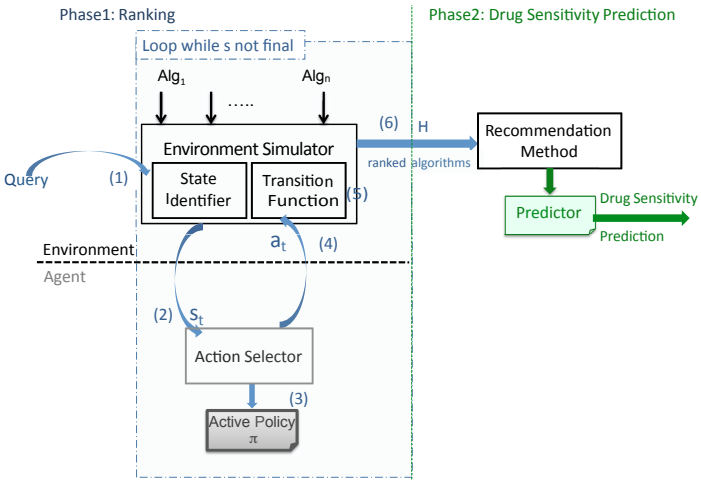

Fig. 4. Prediction mode

i. **Phase 1: Ranking.** In the first phase, Q-Rank applies RL paradigms to find the optimal ranking policy. When the system receives a prediction query, the following process is applied. This process follows six steps (see Figure 4). In Step (1), Q-Rank identifies the state. A state is an observation of the environment that is represented as meta-features (i.e., the types of omics characterizations, the drug) combined with subset of integrated algorithms. In Step (2), the state is sent to the agent. In Step (3), the agent uses the active policy to choose a prediction algorithm. In Step (4), the selected algorithm is sent to the environment, and the algorithm is placed in the ranking list. In Step (5), if the state is not final, the environment transits to next state by removing the selected algorithm from the list of candidate prediction algorithms and resumes from Step 2. Finally, in Step (6), the output is a ranked list of prediction algorithms.

ii. **Phase 2: Drug Sensitivity Prediction.** This phase allows us to compute the sensitivity of the considered cell line(s) to a given drug. It is calculated via the use of a configurable recommendation method. The recommendation method is the approach used to predict drug responses using the ranked list produced during the first phase (i.e., ranking phase). By default, the best ranked algorithm is recommended. It should be pointed out that the sensitivity could be based on all or some of the ranked algorithms (e.g., weighted combination of the sensitivity results), depending on the used approach.

V. Implementation

Q-Rank is implemented using the R programming language and freely available at https://github.com/salma2018/Q-Rank.

A. Data Set

The NCI-DREAM7 data set includes 6 profiled omics characterizations for 53 breast cancer cell lines. These omics profiles include DNA copy number variation (CNV), DNA methylation, and point mutations, transcript expression, RNA sequencing, and protein abundance, as shown in Table I. Drug sensitivity was measured using GI50 ($log_{10}(GI50)$, where GI50 is the drug concentration required to inhibit 50% of maximal cell growth), in response to 28 anti-cancer therapeutic compounds. The unavailable pre-clinical responses GI50 were replaced by NA values. Also, the drug SDF formats were downloaded from the NCBI PubChem Repository. Then, PubChem fingerprint descriptors were computed using PaDEL [24].

TABLE I
Details of the NCI-DREAM7 data set

Total number of samples	53 brest cancer cell lines	
Drug response data	28 compounds	
Omics characterization	**Number of samples**	**Dimension**
RNA sequencing	44 cell lines	36953 transcripts
Gene expression	46 cell lines	18632 genes
SNPs6	47 cell lines	27234 SNPs
DNA methylation	41 cell lines	27551 CpGs7
RPPA	42 cell lines	131 proteins

B. Prediction Algorithms

We used the following prediction algorithms in Q-Rank.

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

1) KBMTL [8]: is a Bayesian formulation that combines kernel-based nonlinear dimensionality reduction and regression in a multitask learning framework in order to solve tasks jointly and improve performance. For this purpose, multiple views of the omic datasets were generated and predictions were made using a kernelized regression method that combines multitask and multiview learning, and uses Bayesian inference to estimate model parameters.

2) IntegratedMRF [9]: is a random forest framework that integrates predictions from different data types. IntegratedMRF generates a multivariate random forest model for each genetic characterization and predicts sensitivity using the generated models. Features are randomly selected to build a set of regression trees for each data set. Then, the model is generated based on averaging the predictions over the collection of trees. The final predictions are calculated using a weighted sum of the individual models.

3) SRMF [10]: is a Similarity-Regularized Matrix Factorization method for drug response prediction that incorporates similarities of drugs and cell lines.

VI. Experimental Results

To evaluate Q-Rank, we use (i) four omics characterizations (i.e., RPPA, GE, ME, SNP), and (ii) the response values to 28 drugs. Also, all combinations (i.e., 15 combinations) of omic characterizations for 53 cell lines are used. Unavailable data is processed as missing data and will impact the performance of the integrated algorithm. Therefore, the robustness of each algorithm against missing data will impact our policy to choose the adequate model. The drug similarities are used by the SRMF method. The matrix of drug sensitivity responses for the cell lines is clustered by drug to sensitive or resistant, using the unsupervised clustering method PAM [25]. Then, the cut-offs are computed. Data normalization is performed using the authors' recommendations for each integrated algorithm. Each integrated algorithm predicts the drug sensitivity as $-log_{10}(GI50)$ values. Then, the predictions are clustered as sensitive or resistant, using the previously calculated cut-offs.

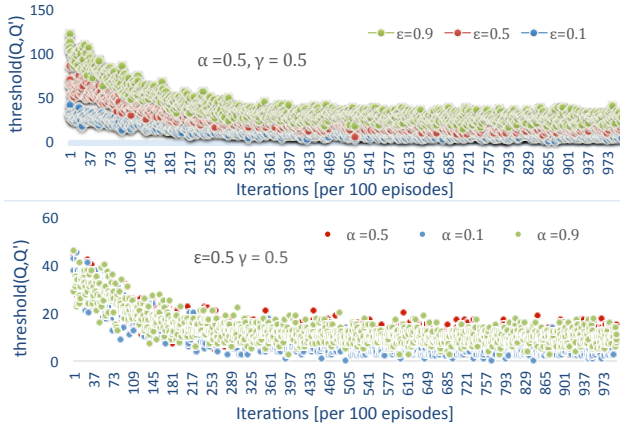

Fig. 5. Sensitivity analysis: Evaluating hyper-parameters to find the optimal policy: the evaluation is based on the difference of (Q,Q') between two iterations. Upper part: variating epsilon for a fixed alpha and gamma. Lower part: variating alpha for a fixed epsilon and gamma.

To find the optimal Q-Rank policy, the learning parameters (i.e., the learning rate $alpha$, the discount rate $gamma$, and the exploration rate $epsilon$) are analyzed. $Gamma$ scales the Q value taken for the next best action. If $gamma$ is reduced, then the immediate reward has more weight. Conversely, if a high gamma value is chosen, the future rewards obtained from the next ranked algorithms are highly considered. Therefore, the $gamma$ value should add value to future rewards but give more value to the immediate reward (first ranked algorithm), which leads to an ideally ranked list. Figure 5 presents a sensitivity analysis of Q-Rank based on $epsilon$ and $alpha$ with a fixed $gamma = 0,5$. With high $epsilon$ values, we are randomly selecting actions. As we reduce epsilon, we select actions more and more greedily improving the results while still ensuring we can explore other actions. With a large $alpha$ value, Q values are learned quickly but might oscillate. A small $alpha$ value will converge more steadily. Hence, the final parameters are $epsilon = 0,1$ and $alpha = 0,1$.

Two set of experiments were conducted: In the first set of experiments, the NCI-DREAM7 data set is divided into 2 groups. The first one includes 35 cell lines. They are used during the "exploration" mode to find the active policy. These cells are also used for training the prediction algorithms. The other 18 cells are used for evaluation during the "prediction" mode.

Q-Rank is evaluated in terms of accuracy, precision, recall, and F1-measure. Furthermore, we provided the Receiver Operating Characteristic curve (ROC curve) which characterizes the sensitivity/specificity tradeoffs of each method. The sensitivity (i.e., true positive rate) is the fraction of sensitive responses (cell, drug), predicted as sensitive. The anti-specificity (i.e., false positive rate) presents the fraction of resistant (cell, drug) pairs classified as sensitive.

For this purpose, Q-Rank is compared to prediction algorithms and competitive approaches that rank existing models. Actually, the integrated algorithms are considered as top-performing methods in previous assessment studies [4], [5]. Therefore, comparing Q-Rank to these algorithms is highly indicative to show that our method outperforms the other prediction algorithms assessed by these studies. Next, Q-Rank is compared to MCRM4Pred, i.e., a method for a per instance algorithm recommendation in drug sensitivity prediction.

Figure 6 shows a comparison of the prediction accuracy between Q-Rank and the other methods. The results show that Q-Rank outperforms KBMTL, MCRM4Pred, and IntegratedMRF by improvements of approximately 5%, 5%, and 8%, respectively.

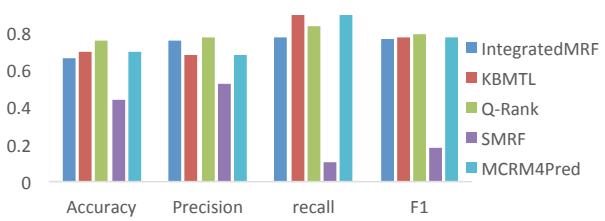

Fig. 6. Q-Rank Vs. other methods

Figure 7 presents the ROC curve, showing that Q-Rank

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

achieves better results than MCRM4Pred and all integrated algorithms, in terms of predicting cell line sensitivity.

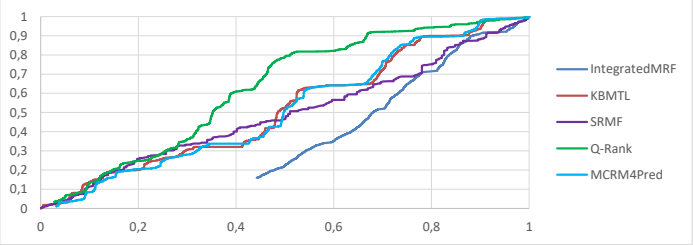

Fig. 7. ROC curves for candidate methods

Furthermore, Q-Rank has been evaluated on the basis of the following factors: (1) the drug, and (2) the type of omics characterization of predicted cell lines.

A. Drug-based Evaluation of Q-Rank

In this evaluation, experimental results are clustered and analyzed on the basis of a drug. As shown in Figure 8, our analysis results demonstrate that the performance indicators of Q-Rank are very close to the best method and occasionally better than all tested methods. On the average, Q-Rank outperforms the integrated algorithms and MCRM4Pred in terms of accuracy, precision, recall, and F1-measure.

Fig. 8. Q-Ranks vs. other methods (drug-based comparison)

B. Omics-based Evaluation of Q-Rank

In our second evaluation scenario, experimental results are clustered and analyzed on the basis of the omic characterizations. Each group is compared to MCRM4Pred and the integrated algorithms. As shown in Figure 9, the analysis results demonstrate that the performance indicators of Q-Rank are mostly better than other methods. On the average, Q-Rank outperforms MCRM4Pred, KBMTL, IntegratedMRF and SRMF in terms of accuracy, precision, recall, and F1-measure.

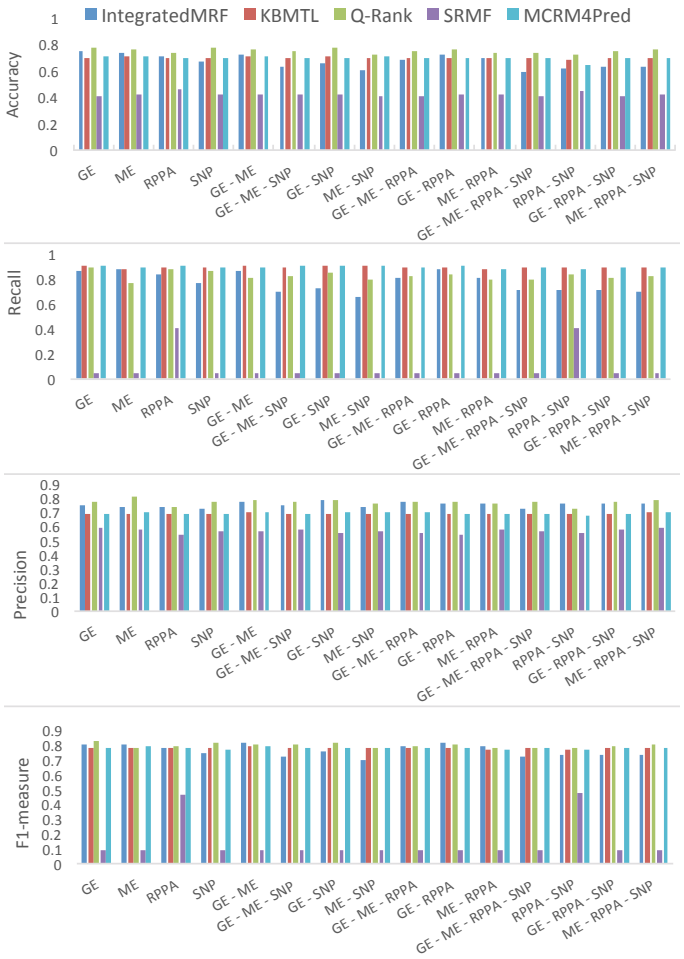

Fig. 9. Q-Ranks vs. other methods (omic-based comparison)

The second set of experiments is performed using 5-fold cross validation to evaluate the merits of Q-Ranks in terms of accuracy. Figure 10 presents comparison results of different methods obtained under 5-fold cross-validation on the NCI-DREAM7 data set. The results confirm that the accuracy of Q-Rank is better than that of other integrated methods.

Fig. 10. Q-Rank vs. other methods (5-fold cross validation)

VII. Discussion

Q-Rank combines existing prediction algorithms in order to predict the sensitivity of anti-cancer drugs. The basic idea is

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2020.3004663, IEEE Journal of Biomedical and Health Informatics

that prediction model performance depends on many factors. Hence, RL is applied to find the optimal ranking of prediction algorithms based on their performance for each case. Then, a recommendation method is applied on the ranked list which defines what prediction model(s) to use according to the ranked list. The default recommendation method selects the first ranked algorithm and uses it to predict the drug responses. It should be pointed out that Q-Rank allows long-life learning by policy iteration on new [pre]clinical data. Experimental results including 5-fold cross validation demonstrate that combining several models and applying each for specific cases can improve the prediction performance. The results show that the prediction models could be complementary, confirming the wisdom of crowd theory. Our evaluation confirms previous results that indicate that (1) GE data is the most informative data and (2) different types of omic data influence the prediction accuracy of algorithms differently which impact the algorithm ranking. However, Q-Rank suffers from some drawbacks. Actually, Q-learning is presented as a tabular mapping of discrete inputs and outputs. This is a limiting factor for continuous states or a large number of states. As the number of integrated algorithms and the number of meta-features increases, the number of states will exponentially increase. Therefore, a more scalabale method is necessary for a larger number of states. In this respect, deep reinforcement learning uses function approximation, as opposed to tabular functions. Specifically, deep reinforcement learning uses deep neural networks to approximate Q-values.

VIII. Conclusion

We presented a novel approach for drug sensitivity prediction, called Q-Rank. It makes use of reinforcement learning techniques to rank prediction algorithms and apply the most suitable algorithm on the basis of the data type and a set of given parameters (i.e., used drug and omics characterization). Our approach was validated in a set of experiments. They demonstrate the merits of Q-Rank when compared to related approaches in terms of precision, recall, accuracy, F1-measure, and ROC curve. Furthermore, these results were confirmed via k-fold cross-validation.

There are several directions for future research. For example, deep learning techniques could be used to deal with scalability issues. Furthermore, it might be beneficial to predict the cell line sensitivity of drug combination [26] by integrating other models and meta-features.

Acknowledgment

This work is supported by the German Academic Exchange Service (DAAD) (Transformation Partnership: Theralytics Project).

References

[1] F. Azuaje, "Computational models for predicting drug responses in cancer research," *Briefings in Bioinformatics*, vol. 18, no. 5, pp. 820–829, 2016.

[2] J. S. Boehm and T. R. Golub, "An ecosystem of cancer cell line factories to support a cancer dependency map," *Nature Reviews Genetics*, vol. 16, no. 7, pp. 373–374, 2015.

[3] I. S. Jang, E. C. Neto, J. Guinney, S. H. Friend, and A. A. Margolin, "Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data," in *Biocomputing 2014*. World Scientific, 2014, pp. 63–74.

[4] Costello, James C et al., "A community effort to assess and improve drug sensitivity prediction algorithms," *Nature Biotechnology*, vol. 32, no. 12, pp. 1202–1212, 2014.

[5] J. Chen and L. Zhang, "A survey and systematic assessment of computational methods for drug response prediction," *Briefings in Bioinformatics*, p. 697896, 2020.

[6] S. Daoud, A. Mdhaffar, B. Freisleben, and M. Jmaiel, "A multi-criteria decision making approach for predicting cancer cell sensitivity to drugs," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM, 2018, pp. 47–53.

[7] M. Fallahi-Sichani, S. Honarnejad, L. M. Heiser, J. W. Gray, and P. K. Sorger, "Metrics other than potency reveal systematic variation in responses to cancer drugs," *Nature Chemical Biology*, vol. 9, no. 11, pp. 708–714, 2013.

[8] M. Gönen and A. A. Margolin, "Drug susceptibility prediction against a panel of drugs using kernelized bayesian multitask learning," *Bioinformatics*, vol. 30, no. 17, pp. i556–i563, 2014.

[9] R. Rahman, J. Otridge, and R. Pal, "IntegratedMRF: random forest-based framework for integrating prediction from different data types," *Bioinformatics*, vol. 33, no. 9, pp. 1407–1410, 2017.

[10] L. Wang, X. Li, L. Zhang, and Q. Gao, "Improved anticancer drug response prediction in cell lines using matrix factorization with similarity regularization," *BMC Cancer*, vol. 17, no. 1, p. 513, 2017.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Bradford Books, 2018.

[12] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement learning*. Springer, 2012, pp. 45–73.

[13] A. Bonarini, C. Caccia, A. Lazaric, and M. Restelli, "Batch reinforcement learning for controlling a mobile wheeled pendulum robot," in *IFIP International Conference on Artificial Intelligence in Theory and Practice*. Springer, 2008, pp. 151–160.

[14] P. Geeleher, N. J. Cox, and R. S. Huang, "Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines," *Genome Biology*, vol. 15, no. 3, p. R47, 2014.

[15] M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester, and J. Saez-Rodriguez, "Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties," *PLoS One*, vol. 8, no. 4, p. e61318, 2013.

[16] C. Huang, R. Mezencev, J. F. McDonald, and F. Vannberg, "Open source machine-learning algorithms for the prediction of optimal cancer drug therapies," *PLoS One*, vol. 12, no. 10, p. e0186906, 2017.

[17] Z. Dong, N. Zhang, C. Li, H. Wang, Y. Fang, J. Wang, and X. Zheng, "Anticancer drug sensitivity prediction in cell lines from baseline gene expression through recursive feature selection," *BMC Cancer*, vol. 15, no. 1, p. 489, 2015.

[18] N.-N. Guan, Y. Zhao, C.-C. Wang, J.-Q. Li, X. Chen, and X. Piao, "Anticancer drug response prediction in cell lines using weighted graph regularized matrix factorization," *Molecular Therapy-Nucleic Acids*, vol. 17, pp. 164–174, 2019.

[19] R. Rahman, K. Matlock, S. Ghosh, and R. Pal, "Heterogeneity aware random forest for drug sensitivity prediction," *Scientific Reports*, vol. 7, no. 1, pp. 1–11, 2017.

[20] H. Liu, Y. Zhao, L. Zhang, and X. Chen, "Anti-cancer drug response prediction using neighbor-based collaborative filtering with global effect removal," *Molecular Therapy-Nucleic Acids*, vol. 13, pp. 303–311, 2018.

[21] L. Zhang, X. Chen, N.-N. Guan, H. Liu, and J.-Q. Li, "A hybrid interpolation weighted collaborative filtering method for anti-cancer drug response prediction," *Frontiers in Pharmacology*, vol. 9, p. 1017, 2018.

[22] C. De Niz, R. Rahman, X. Zhao, and R. Pal, "Algorithms for drug sensitivity prediction," *Algorithms*, vol. 9, no. 4, pp. 77–101, 2016.

[23] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[24] C. W. Yap, "Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints," *Journal of Computational Chemistry*, vol. 32, no. 7, pp. 1466–1474, 2011.

[25] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, M. Studer *et al.*, "Package cluster," *Dosegljivo na*, 2013.

[26] X. Chen, B. Ren, M. Chen, Q. Wang, L. Zhang, and G. Yan, "NLLSS: predicting synergistic drug combinations based on semi-supervised learning," *PLoS Computational Biology*, vol. 12, no. 7, p. e1004975, 2016.

2168-2194 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Exeter. Downloaded on June 26,2020 at 14:27:27 UTC from IEEE Xplore. Restrictions apply.