Reconfigurable Probabilistic AI Architecture for Personalized Cancer Treatment

Sourabh Kulkarni, Sachin Bhat, Csaba Andras Moritz
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA
skulkarni@umass.edu, andras@ecs.umass.edu

***Abstract*— The machinery of life operates on the complex interactions between genes and proteins. Attempts to capture these interactions have culminated into the study of Genetic Networks. Genetic defects lead to erroneous interactions, which in turn lead to diseases. For personalized treatment of these diseases, a careful analysis of Genetic Networks and a patient's genetic data is required. In this work, we co-design a novel probabilistic AI model along with a reconfigurable architecture to enable personalized treatment for cancer patients. This approach enables a cost-effective and scalable solution for widespread use of personalized medicine. Our model offers interpretability and realistic confidences in its predictions, which is essential for medical applications. The resulting personalized inference on a dataset of 3k patients agrees with doctor's treatment choices in 80% of the cases. The other cases are diverging from the universal guideline, enabling individualized treatment options based on genetic data. Our architecture is validated on a hybrid SoC-FPGA platform which performs 25x faster than software, implemented on a 16-core Xeon workstation, while consuming 25x less power.**

Keywords— Computer Architecture, Reconfigurable Computing, SoC-FPGA, Probabilistic Computing, Bayesian Networks, Personalized Medicine, Interpretable AI, Genomics.

I. INTRODUCTION

This work focuses on the problem of discovering gene networks (GNs) and their application in determining treatment choices for breast cancer patients. Globally, breast cancer is the most common cancer in women with close to 1.7 million cases diagnosed annually. The average 5-year survival rate of stage 1 (local) breast cancer is 98.5% but drops significantly to 84% and 24% for stage 2 (regional) and stage 3 (distant), respectively[1]. Hence, an early diagnosis and treatment of breast cancer is crucial, which this work aims to impact. Decades of breast cancer research has culminated into a deep understanding of its various subtypes and has led to the formation of various GNs for breast cancer[2][3]. GNs, as shown in Fig. 1, are a compact representation of our understanding of the *genetic basis of diseases*[4]. Recent developments indicate that *probabilistic* AI models may be used to capture GN structure[5]. These AI models, such as Bayesian Networks (BNs), are based on probability theory and Bayesian statistics. They intrinsically support asking questions such as those in a personalized medicine context. They represent knowledge in an interpretable graph and can provide realistic confidences in their predictions, which is important for medical applications. Once a GN structure is discovered and modeled using probabilistic AI, it can be used to predict the likelihood that a certain patient has breast cancer based on their gene expression profile. Furthermore, it can also be used to infer the best possible treatment for a patient already diagnosed of cancer.

Currently, this process is time-consuming, and expensive. A state-of-the-art algorithm performing a GN-based benchmark with 3000+ genes requires ~325 Xeon processors and ~170 hours of runtime[6]. This is because, implementing probabilistic AI in software involves stochastic computations over several layers of abstraction, on circuits and architectures that are deterministic in nature. Alternatively, personalized treatment choices are made by expert oncologists at a per-patient basis. Currently, only few institutions in the world have access to such enormous computing resources and expertise. Hence, for most cancer patients, the treatment choice is not personalized, but is generalized to factors such as age, gender, stage of cancer etc. These limitations can be overcome by designing architectures that support these probabilistic AI models where compute-intensive operations are significantly accelerated.

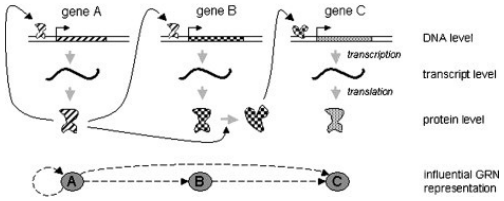
In this work, we co-design a probabilistic AI model and reconfigurable architecture for (i) Discovering GN structures by modelling them as BNs; and (ii) Performing inference on these BNs for personalized-treatment selection. To enable these operations, we develop modified, hardware-aware versions of BN learning and inference operations. These are implemented on an architecture that maps BN graphs onto a reconfigurable FPGA fabric that consists of several Stochastic Bayesian Nodes and supports arbitrary connectivity among them. The reconfigurability allows incorporating new domain expertise by supporting modifications to GNs, while the stochasticity of nodes allows for efficient probabilistic computations. We utilize gene expression data of ~3k breast cancer patients for validating our approach on an SoC-FPGA platform. This approach can be extended to other diseases and medical applications for humans and other life forms.

The paper is organized as follows: Section II provides a background on Bayesian Networks, the probabilistic AI models used in this work. We then describe the problem statement in section III. Section IV consists of the architectural and implementation details. Section V discusses the results and related analysis. Section VI concludes the paper.

II. BACKGROUND

Bayesian Networks (BNs) are graph-based probabilistic AI models that attempt to capture knowledge of a domain by encoding the various entities as random variables and the relationships between them as causal connections. The strength of causal relation between variables is represented as a

(A)



(B)

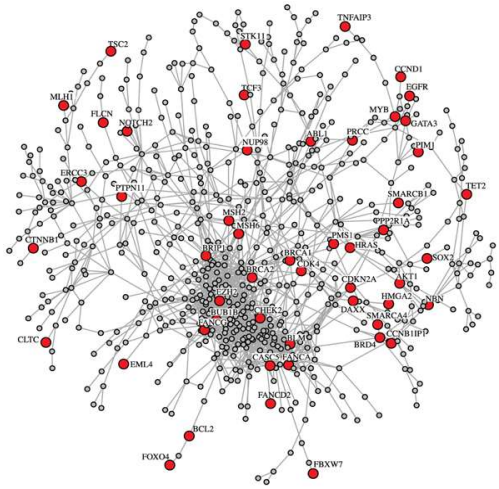


Fig. 1. (A) Schematic of a simple gene network[9] portraying gene-gene interaction through transcription and protein synthesis. (B) Subnetwork of the genome-scale breast cancer GN (19k genes); part consists of 51 breast cancer genes, 677 related genes and their interactions[10].

conditional probability table (CPT). The two major aspects for using BN models is learning and inference. Both operations are of high computational complexity and scale exponentially in the worst case[7][8].

Learning: The two major aspects involved in learning a BN model are its graph structure and its CPT parameters. The structure of the BN is inferred from data through combinatorial optimization techniques or constructed through domain experts' knowledge. This work employs a combination of both of these techniques. For learning the CPT values, we use the maximum likelihood estimate (MLE) method, which is quite suitable for our architecture and is one of the most widely used.

Inference: BN model inference can be either analytic or sampling based. Analytic inference methods tend to be exact, but are compute intensive, while sampling-based methods are approximate but potentially more computationally feasible, depending on platform. In this work, the sampling-based method is used, as it can be massively parallelized in hardware and is amenable to modifications in its implementation. These properties are useful in design of a hardware-aware inference method for BNs which can be implemented by simple computational elements.

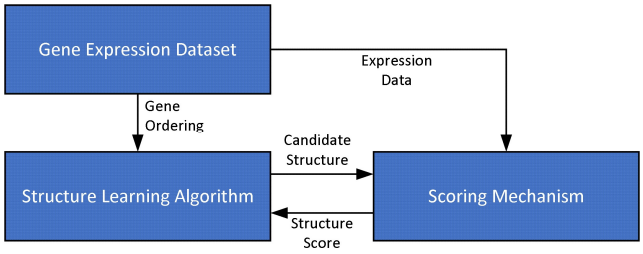


Fig. 2. Design flow for structure learning in BNs. Learning algorithm generates graphs from data, which are scored by the scoring mechanism. The algorithm iteratively improves upon the graph structure until structure scores over a certain threshold.

III. Probabilistic AI Model

This section discusses the learning and inference of the probabilistic AI model for personalized treatment. Here, GNs whose structure is to be discovered are modelled as BNs; each gene in the GN becomes a node in the BN. Hence, the problem of structure discovery for GNs is mapped to the problem of finding the best possible connections between nodes of a BN. Fig. 2 shows a high-level view of this process. The structure learning algorithm comes up with candidate structures, which are scored by a scoring mechanism. Candidate structures are generated by incrementally building on smaller structures. The scoring mechanism determines how well the candidate structure explains the genetic data. The scoring mechanism computes the *posterior structure probability* $P(structure|data)$ using the Bayes' rule as follows:

$$P(structure|data) \propto P(data|structure) * P(structure) \quad (1)$$

Where $P(data|structure)$ is the *structure likelihood* of the structure while $P(structure)$ is the *structure prior*. *Structure prior* incorporates the order in which genes are added to the structure and can be obtained from literature[3][11]. *Structure likelihood* is computed by averaging over *individual node likelihoods* in the structure:

$$P(data|structure) = \frac{1}{N}\sum_{i\in N} P(node_i, data|structure) \quad (2)$$

The *individual node likelihoods* are independent of the rest of the structure given the *parent* state, hence are approximated by performing local inference over the node and its parent using Gibbs sampling method:

$$P(node_i, data|structure) \approx 1 - \left(\frac{1}{S}\sum_{j\in S}\left(P\left(node_i|parentSample_j\right)\right) - P(node_i|data)\right) \quad (3)$$

Here, the first term represents the Gibbs' sampling inference performed over S samples and the second term is the *marginal probability* of the node given the data. The smaller the difference between these terms, the better the node's position

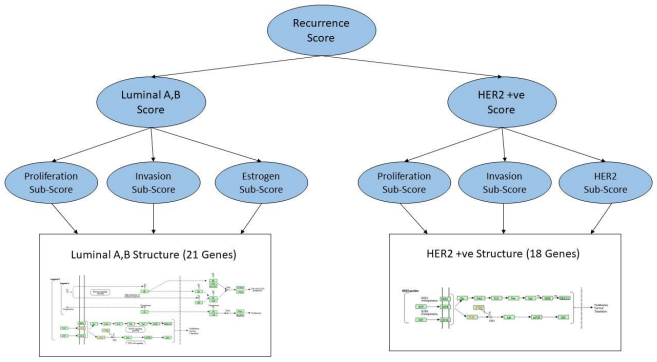


Fig. 3. Hierarchical Bayesian Latent Variable Model designed for computing probability of cancer recurrence. 'Observed' variables (bottom squares) correspond to gene expression values, 'latent' variables (blue nodes) correspond to known sub-scores associated with cancer recurrence.

and data are in agreement with the structure. Although this is an approximation, it converges to the exact result with enough samples. In this case we choose a sample size of 20k to obtain a precision of 0.00005 within the correct answer, which was sufficient for the application.

The nature of this optimization problem prohibits any proof of optimality of the candidate structure; hence, the design process is terminated when the structure is 'good enough', which is denoted by a threshold of loss function. It is therefore possible for several GN structures to agree with genetic data.

The other important aspect of this work is to predict the best possible personalized treatment. This is done by introducing latent variables in the learnt structure. The model takes personalized genetic information of a patient, and the latent models hierarchically compute a 'recurrence score': the probability of the cancer tumor to recur post-treatment. The recurrence score can then be used to select a treatment that is best suitable for the patient. The approach of recurrence-score based treatment choice selection has been validated through clinical trials[12], although by using less complex AI algorithms.

We construct the model, as shown in Fig. 3, that utilizes the GNs like the HER2 network learnt previously and another network which corresponds to Luminal A and B type breast cancers to hierarchically compute a recurrence score for each patient. These intermediate sub-scores cannot be observed from data, hence are called latent (or hidden) variables. Computing the parameters for these latent variables requires specialized analysis of data, known as factor analysis. Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a lower number of unobserved (latent) variables called factors. The factors correspond to the first order of latent variables, which are sub-scores (proliferation, invasion, etc.) related to various subtypes of Breast Cancer. The second-order latent variables encode the probability of recurrence of certain types of Breast Cancer (Luminal A/B, HER2 +ve). The third order latent variable is the total recurrence score - the overall likelihood of recurrence of the cancer tumor. A succinct representation of the hierarchical inference operation we do over the model is as follows:

$$Recurrence\ Score = P(recurrence = high|subscores) * P(subscores|expression\ data) \quad (4)$$

Here, the $subscores$ are inferred from gene $expression\ data$ of genes in GN, while the $Recurrence\ Score$ is inferred from the $subscores$. Each of these computations uses Gibbs' sampling for inference and resembles equation (3). The recurrence score is helpful for doctors to decide on treatment for the patient. A low recurrence score (low probability of recurrence of cancer) can indicate that a less harmful treatment (e.g., endocrine treatment) should suffice. A high recurrence score indicates that a more potent but also potentially more harmful treatment (like, e.g., chemo plus endocrine treatment) may be required.

Dataset: The dataset used in this project is obtained from the Gene Expression Omnibus (GEO) Database[13]. It consists of expression values of 20,000 genes of 3,070 breast cancer patients, along with their medical information – the type of cancer diagnosed, treatments given, and the survival event. To facilitate the structure discovery, it is helpful to refer to prior information from literature. This prior information is related to what subset among the 20k genes to consider (40 genes considered). The data has been discretized into three states *High expression* (H), *Medium expression* (M) and *Low expression* (L). For validating the personalized treatment model, the treatment predictions based on the expression profiles of patients are compared with the actual treatments given to the patients and the effectiveness of those treatments.

IV. System Architecture

From a computational point of view, the probabilistic AI model can be divided into four important steps:

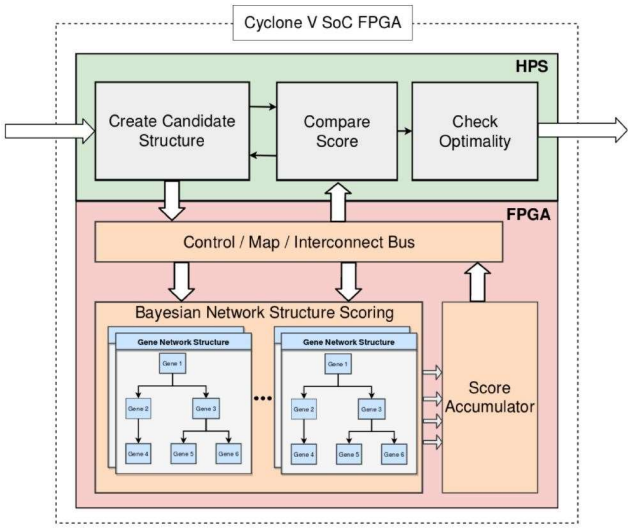


Fig. 4.Overall Architecture for structure learning task. The candidate structures are generated and incremented in software on the HPS and are mapped to FPGA for scoring.

1. *Candidate Structure Generation* – Given the ordering of genes, generate a structure with random connections.
2. *Scoring Candidate Structures* – Multiple inference operations through Gibbs sampling to generate a score for a given structure.
3. *Structure Selection* – Given multiple structures, compare their scores to select the best possible structure given data.
4. *Recurrence Score Computation* – Inference operation on the learnt structure with latent variables based on the gene expression data of a patient.

The computations in the first and third steps have a sequential flow while the second and fourth steps can be massively parallelized. Here, we chose to implement the sequential computations using software on a general-purpose processor and the parallel computations on a customized hardware based on FPGA. The software implements the sequential control flow while the custom hardware significantly accelerates scoring aspects. We chose an SoC-FPGA platform to implement our design as shown in Fig. 4. SoC-FPGAs tightly integrate both processor (HPS) and FPGA architectures on a single die. This configuration allows us to design an architecture where computational loads could be distributed among the two platforms for efficient utilization of both.

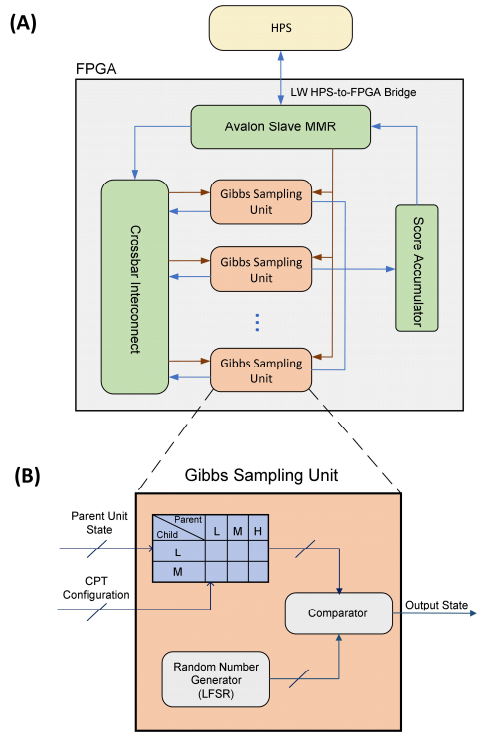


Fig. 5. FPGA-Based Proposed Architecture. (A) Shows connectivity architecture for instantiating a candidate structure in FPGA fabric. Individual nodes are mapped into Gibbs Sampling Units by instantiating the corresponding CPTs, while graph connectivity is programmed through crossbar interconnect; (B) Shows architecture of a Gibbs Sampling Unit. Parent unit state selects an entry in the CPT, which is compared to output of LFSR to determine output state of current unit.

A. Hardware Architecture for Scoring Candidate Structures

Scoring of the candidate structures is implemented on a custom reconfigurable hardware instantiated on the FPGA. The architecture allows for efficient implementation of Gibbs-sampling based inference for scoring candidate structures. The candidate structure to be scored is mapped on the FPGA. The hardware comprises of Gibbs Sampling Units which can be arbitrarily connected to each other through the programmable crossbar interconnect. Each unit consists of a soft-configurable input multiplexer that can be configured to select any other unit in the structure as its parent unit. The system is interfaced to the software using a SOPC wrapper that implements an Avalon slave memory-mapped register (MMR) IP. The software configures the network parameters, such as the input multiplexer configuration, CPT values, number of samples, by writing to the MMRs. Fig. 5(A) shows a detailed block diagram of the proposed architecture.

Gibbs Sampling Unit – A detailed block diagram of the Gibbs Sampling Unit in FPGA is shown in Fig. 5(B). The candidate structure to be scored is mapped onto the framework by mapping its nodes to Gibbs Sampling Units. Each unit consists of registers to store the CPT values with 8-bit precision. The stochastic sampling is achieved by employing a Linear Feedback Shift Register (LFSR) based pseudo-random generator in each node. It is seeded by a unique value to generate an independent sequence of 8-bit random values. A row is selected from the CPT based on the parent nodes state which is then compared with the random value to generate output state. Each output state is a 'sample' in the Gibbs sampling process. These samples are accumulated and sent to the HPS where the structure score is computed according to equation (3).

B. Software Design

The software running on the processor implements the first and third step of the AI model. The algorithm is summarized as follows:

```
BEGIN
READ expressionValues // gene expression data
READ nodeOrdering // structure prior
INITIALIZE candidateStructure with nodeOrdering.firstNode()
APPEND candidateStructure with nodeOrdering.nextNode()
CalculateCPTs()
WHILE (size(candidateStructure) < maxSize):
   APPEND candidateStructure with nodeOrdering.nextNode()
   FOR node in candidateStructure:
      AddAsParentNode()
      CalculateCPTs()
      score = ComputeStructureScore() // done in FPGA
      IF (score >= bestScore):
         KeepAsParentNode()
      ELSE:
         RemoveAsParentNode()
OUPUT bestStructure = candidateStructure
END
```

Candidate Structure Generation: The order in which the genes are added (nodeOrdering) is predetermined from literature. The software follows this order for generating the candidate structures. Initially, the search starts with a single gene in the network. As there are no parent genes to the first gene, the CPT entries for that node corresponds to the marginal probabilities of the three states for that gene, i.e., percentage of patients who have high, medium and low expression values. Subsequently more genes are added. The CPT values for these genes correspond to the probability of a gene being in one of the three states given that the parent gene is in one of the states. Once the CPT values are populated for the candidate structure, it is ready for scoring. The network along with its CPT values are written to the FPGA memory-mapped registers. The FPGA then performs the sampling and returns the score for that network.

Structure Selection: In this step, the network that will produce the best score from each order is determined. This is done by comparing the score of the network to the ground-truth (data). More and more genes are subsequently added to the network and scored. Finally, the structure with the best score is selected.

Recurrence Score Computation: Essentially, the structure discovery process involves performing multiple inferences on the various candidate structures. Hence, the same FPGA architecture can be used to perform inference as well as structure scoring. In the case of recurrence score computation, the structure is already known. The structure along with the CPTs is mapped onto the FPGA. After the sampling, the inference results for recurrence score are written back to the processor memory.

V. Analysis and Results

A. Correctness of Learnt Gene Network

The final structure generated by the algorithm is compared with a reference structure from the *KEGG Gene Pathway* database to validate the approach. Note that such comparison is not expected to be available for all applications, but we choose this specific GN such that we can also prove our design. The KEGG database is a state-of-the-art extensive library which maintains and updates a large collection of reference gene networks. Finding the correct structure for a GN from expression data is a hard problem and is currently an area of active research. As of now, there is no well-established notion of 'correctness' for a GN. As this is a small GN, we could compare it with the pre-existing GN from the KEGG database. The implementations generated are not an exact copy of the KEGG reference GN, as the reference GN is formed by compiling the results of several research efforts and we used currently a more limited dataset. This is, however, not a limitation as the qualitative benefits when scored across a larger dataset would persist. Furthermore, for the sake of simplicity, the structures being learnt in this project were restricted to be trees, which leads to few changes relative to KEGG.

B. Personalized Treatment Inference

We have developed a validation scheme for the model to gauge its accuracy. The validation is done as follows: By using the recurrence score generated by those models, we suggest a treatment. The Patient dataset has data regarding what treatment was provided to each patient along with 5-year survival event (whether patient survived for 5-yrs post treatment). This information is used to validate to what degree the treatment suggested by the model agrees with the doctors. We do a cross-validation of over 3070-patient dataset based on data from multiple hospitals[13]. The treatments provided to the patients did not include a genetic analysis but were made based on clinical factors and the doctor's expertise. As expected, from our validation, the treatments suggested by the doctors and the ones suggested by our inference model did not fully overlap but overlapped around 80% of the time. For 20% of the cases, the treatment choice suggested was different; this is likely because the gene expression-based approach is able to infer a more personalized treatment choice. That could either mean that a stronger treatment was given to patient than necessary, leading to unnecessary side effects, or that the treatment given was not strong enough, leading to sub-optimal treatment. It is to be noted that (in the general sense) it has already been shown in the research community that the gene expression-based approach for inference is more accurate[11][15] than using clinical factors alone.

TABLE 1. CYCLONE V FPGA RESOURCE USAGE FOR LEARNING AND INFERENCE OF 41-NODE GN.

	Usage	**%**
Logic Utilization	17060/41910	41
Total LABs	2570/4191	61
I/O Pins	172/314	55
M10K Blocks	25/553	4.5

C. Prior Related Works

While this work, to the best of our knowledge, is the first attempt for implementing Probabilistic AI Architecture for personalized medicine, there have been some works that focus on implementing BNs on FPGAs[17][18]. These approaches build on producing 'processing units' on the FPGA and distributing the inference task workload. Such an approach leads to resource-heavy architecture which is not scalable. For example, one unit in [17]takes 12-24% of the FPGA area. This requires that designs be partitioned and loaded partially into the hardware several times, hence supporting larger applications would not be feasible. In our work, the individual units are relatively simple stochastic circuits tasked for probability distribution sampling which makes our approach much more scalable.

D. Scalability of Probabilistic AI Architecture

The size of BNs which map GNs vary from few nodes to several tens of thousands of nodes depending on the gene pathway considered. Since the number of nodes that are to be mapped scales linearly with the logic blocks, the application can be easily scaled conditioned upon the availability of the logic blocks on the FPGA. For the breast cancer gene network considered, 14% of the total resources were utilized for 18-node GN including the resources utilized by GHRD that came along with the board, while for a 41-node GN, 41% resources were utilized (See Table 1). We estimate that GNs with 100s of

TABLE 2. PERFORMANCE AND POWER BENEFITS OF FPGA-BASED PROPOSED ARCHITECTURE VS. SOFTWARE BASELINE IN R

	Mean Runtime			Power (W)		
	Baseline	**Proposed**	**Benefit**	**Baseline**	**Proposed**	**Benefit**
Structure Learning	0.6s	0.5s	~1.2x	135W	5W	~25x
Inference	0.05s	0.002s	~25x	135W	5W	~25x

nodes are feasible on the Cyclone V FPGA. If the candidate Bayesian graphs are less than 100 nodes in size, several graphs can be mapped onto the FPGA at once, thus enabling several graphs to be scored concurrently. For the breast cancer network considered, up to 5x speedup is possible for the 18-node GN by scoring 5x candidate structures in parallel. Similarly, for inference, the same GN can be instantiated multiple times to obtain a performance gain of 3-5x (vs. 1.2x for a single GN instantiation). The inference GN model was replicated thrice in the FPGA to better utilize the resources of the FPGA. The FPGA utilization resources for the structure learning and inference models are given below in the Table 1.

E. Performance Evaluation and Comparison with Software Baseline

Baseline: We compare our results with a similar implementation in R, which is a data-analysis programming language widely used in the bioinformatics domain. R has several libraries dedicated to Bayesian structure discovery, of which '*bnlearn'* was chosen. The hill-climbing algorithm (HC) was used for structure discovery. This algorithm was executed on a 4GHz, 8-core/16-thread Xeon workstation. The runtime for structure discovery operation using this configuration is listed in the table below. Performance benefits will be greater for larger networks, as even high-end CPUs are limited by core count, while the FPGA implementation of the score operation is not. The runtimes for larger network sizes will increase in CPUs while remaining near constant on proposed architecture.

Upon comparisons with the software-based approach, we observed around 25x improvement in the per-patient inference time as shown in Table 2. We estimate additional 3-5x improvement in both learning and inference if multiple GN models are instantiated on the FPGA in parallel. This is because some of the resources can be shared by multiple GN model instantiations. While the entire design with just one GN model takes up ~41% (see Table 1), we estimate that we can fit up to 5 GN models to achieve full utilization of the FPGA. The human genome consists of over 20k genes. We expect that not only the performance will improve with the size of the network but also the type of problem that we could solve would go way beyond what we have shown. Acceleration in learning and inference with up to two orders of magnitude, which we demonstrated, would allow us to consider much larger gene networks and expand this system to a wide range of other diseases.

VI. CONCLUSION

We design a probabilistic AI model for suggesting personalized treatment options for cancer patients. Furthermore, we develop a reconfigurable architecture for implementing this model where the computational workload is distributed between software and FPGA. In our model, Gene Networks are modeled as Bayesian networks for structure discovery of Breast Cancer GNs from a Gene Expression dataset of ~3k patients. We provide interpretable and personalized treatment options to Breast Cancer patients based on the latent variable model we designed for utilizing the learnt GNs. Our model agrees with the doctor's treatment choices 80% of the times, while the departure in the rest of the treatment choices could be attributed to the personalized nature of the model. Several research efforts[11][12] have shown that use of personalized genetic information leads to better diagnosis and treatment of cancer patients. To that end, we are currently in collaboration with oncologists and genetic researchers from University of Nebraska Department of Genetics and Cell Biology, and University of Debrecen Medical School, Hungary toward verifying the effectiveness of treatment choices of our model. The architecture implemented on a Cyclone V SoC-FPGA performs 25x faster and is 25x more power efficient that software-only implementation on a Xeon workstation. Our prototype platform demonstrates the feasibility of realizing a relatively low-cost solution for targeting key applications in personalized medicine. Our ultimate vision is a personalized medicine system which can be made available in hospitals and clinics all over the world. Such system would be able to perform at low cost and high-performance Bayesian inferences towards interpretable and personalized diagnosis and treatment of cancer patients. The system's reconfigurability would allow support for multiple healthcare solutions and the ability to assimilate the latest breakthroughs in medicine toward improved effectiveness.

REFERENCES

[1] Breast Cancer Research Fondation, "Breast Cancer Statistics," Available: https://www.bcrf.org/breast-cancer-statistics

[2] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, "Gene regulatory network inference: Data integration in dynamic models-A review," BioSystems, vol. 96, no. 1, pp. 86–103, 2009.

[3] Kyoto Encyclopedia of Genes and Genomes, "Breast cancer - Homo sapiens (human)," Available: https://www.genome.jp/kegg-bin/show_pathway?hsa05224=

[4] W. K. Cavenee, and R. L. White, "The genetic basis of cancer," Scientific American, vol. 272, no. 3, Mar., pp. 72-79, 1995.

[5] J. M. Pena, "Learning and Validating Bayesian Network Models of Genetic Regulatory Networks," Advances in Probabilistic Graphical Models, vol. 214, pp. 359–375, 2007.

[6] A. Frolova and B. Wilczyński, "Distributed Bayesian networks reconstruction on the whole genome scale," PeerJ, vol. 6, p. e5692, 2018.

[7] Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. Artificial intelligence, 42(2-3), 393-405.

[8] Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. Artificial intelligence, 60(1), 141-153.

[9] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, "Gene regulatory network inference: Data integration in dynamic models-A review," *BioSystems*, vol. 96, no. 1, pp. 86–103, 2009.

[10] F. Emmert-Streib, R.M. Simoes, P. Mullan, B. Haibe-Kains, and M. Dehmer, "The gene regulatory network for breast cancer: integrated regulatory landscape of cancer hallmarks," Frontiers in genetics, vol. 5, 2014. Available: 10.3389/fgene.2014.00015

[11] M. L. McCullough, R. M. Bostick, and T. L. Mayo, "Vitamin D gene pathway polymorphisms and risk of colorectal, breast, and prostate cancer," Annual review of nutrition, vol. 29, pp. 111-132, 2009.

[12] J. A. Sparano, R. J Gray, D. F. Makower et al., "Adjuvant Chemotherapy Guided by a 21-Gene Expression Assay in Breast Cancer," New England Journal of Medicine, vol. 379, pp. 111–121, 2018.

[13] National Center for Biotechnology Information, "Gene Expression Omnibus," Available: https://www.ncbi.nlm.nih.gov/geo/

[14] T. O'Mara, D. Duffy, D. Thompson, et al., "A genome-wide association study to identify genetic markers associated with endometrial cancer grade," Hered Cancer Clin Pract, vol. 10, no. S2, p. A47, 2012.

[15] M. J. Garnett, E. J. Edelman, et al., "Systematic identification of genomic markers of drug sensitivity in cancer cells," Nature, vol. 483, no. 7391, pp. 570-575, 2012.

[16] Kyoto Encyclopedia of Genes and Genomes, "Breast cancer - Homo sapiens (human)," Available: https://www.genome.jp/kegg-bin/show_pathway?hsa05224

[17] Z. Kulesza and W. Tylman, "Implementation Of Bayesian Network In FPGA Circuit," Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006., Gdynia, 2006, pp. 711-715.

[18] M. Lin, I. Lebedev, and J. Wawrzynek, "High-Throughput Bayesian Computing Machine with Reconfigurable Hardware," Proc. Int. Symp. F. Program. Gate Arrays, pp. 73–82, 2010.