

Structural bioinformatics

# GPU-I-TASSER: a GPU accelerated I-TASSER protein structure prediction tool

Elijah A. MacCarthy<sup>1</sup>, Chengxin Zhang <sup>2</sup>, Yang Zhang<sup>2,\*</sup> and Dukka B. KC <sup>3,\*</sup>

<sup>1</sup>Department of Mathematics, Lane College, Jackson, TN 38301, USA, <sup>2</sup>Department of Computational Medicine & Bioinformatics, University of Michigan, Ann Arbor, MI 48109, USA and <sup>3</sup>Computer Science Department, Michigan Technological University, Houghton, MI 49931, USA

\*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on September 17, 2021; revised on December 18, 2021; editorial decision on December 27, 2021; accepted on December 28, 2021

## Abstract

**Motivation:** Accurate and efficient predictions of protein structures play an important role in understanding their functions. Iterative Threading Assembly Refinement (I-TASSER) is one of the most successful and widely used protein structure prediction methods in the recent community-wide CASP experiments. Yet, the computational efficiency of I-TASSER is one of the limiting factors that prevent its application for large-scale structure modeling.

**Results:** We present I-TASSER for Graphics Processing Units (GPU-I-TASSER), a GPU accelerated I-TASSER protein structure prediction tool for fast and accurate protein structure prediction. Our implementation is based on OpenACC parallelization of the replica-exchange Monte Carlo simulations to enhance the speed of I-TASSER by extending its capabilities to the GPU architecture. On a benchmark dataset of 71 protein structures, GPU-I-TASSER achieves on average a 10× speedup with comparable structure prediction accuracy compared to the CPU version of the I-TASSER.

**Availability and implementation:** The complete source code for GPU-I-TASSER can be downloaded and used without restriction from <https://zhanggroup.org/GPU-I-TASSER/>.

**Contact:** zhng@umich.edu or dbkc@mtu.edu

**Supplementary information:** [Supplementary data](#) are available at Bioinformatics online.

## 1 Introduction

The development of computational methods to accurately model protein structures from sequences is one of the important problems in the field of structural bioinformatics. Recently, there have been various advances in the field of protein structure prediction as witnessed in the community-wide CASP experiments (Kryshtafovych *et al.*, 2019, 2021), in which several excellent methods have stood out, including Iterative Threading Assembly Refinement (I-TASSER) (Yang *et al.*, 2015), Rosetta (Rohl *et al.*, 2004), QUARK (Xu and Zhang, 2012), RaptorX (Kallberg *et al.*, 2014) and most recently AlphaFold (Jumper *et al.*, 2021). In I-TASSER, the modeling process starts with template identification from the PDB, where the full-length structure models are constructed by reassembling structural fragments from template alignments using Replica Exchange Monte Carlo (REMC), which performs concurrent simulations of  $n$  ( $=40$ ) different replicas of the MC system, each running under different temperatures.

REMC is the most time-consuming process in the I-TASSER pipeline and this step becomes the bottleneck particularly for large-size proteins. Therefore, we have developed a very fast version of I-TASSER for Graphics Processing Units (GPU-I-TASSER) by implementing a parallel execution of the REMC method. Essentially, we develop GPU-

I-TASSER by porting compute-intensive operations to the GPU using OpenACC. The availability of GPU-I-TASSER is likely to facilitate the widespread adoption of GPU-I-TASSER by the structural bioinformatics community. It will complement the existing end-to-end protein structure prediction packages such as AlphaFold2 (Jumper *et al.*, 2021) and RoseTTAFold (Baek *et al.*, 2021), especially when restraints from multiple structural templates need to be imposed for structure prediction.

## 2 Materials and methods

The I-TASSER pipeline for the protein structure prediction consists of three main steps: threading template identification, iterative REMC structure assembly simulation, and clustering and structure refinement (Yang *et al.*, 2015). REMC within structure assembly consists of expensive energy calculation and Monte Carlo moves/updates and swaps that take up approximately 75% of the total computational time. As a result, this step becomes the ideal target for porting the expensive calculation to the GPU. GPUs accelerate calculations by executing the same instructions simultaneously on different data (i.e. SIMD paradigm). For different replicas, which represent (multiple) different data, the same operations (single

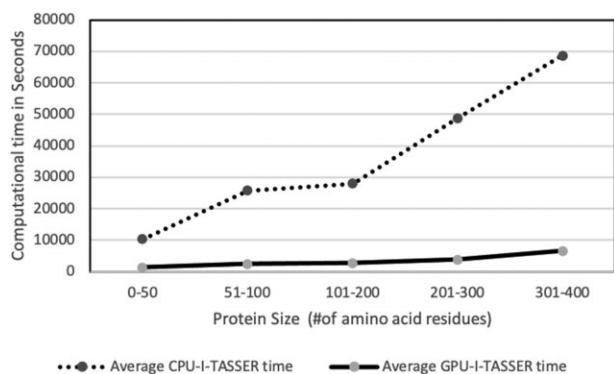


Fig. 1. Average execution time (in seconds) for GPU-I-TASSER and CPU-I-TASSER (original I-TASSER) for various protein sizes binned according to the number of protein residues

instruction) of energy calculations, Monte Carlo moves and swaps need to be performed. Reduction kernels are also used on operations that involve reductions with results synchronized. We describe in detail the porting steps below.

### 2.1 Energy computations

The knowledge-based force field of I-TASSER consists of three major components: (i) general statistical potential, (ii) hydrogen-bonding energy and (iii) threading-based restraints from LOMETS (Zheng *et al.*, 2019). Computations on these three force field components take a substantial amount of time. Hence, we tackle each of the components on the GPU and utilize the massive number of threads to facilitate the computational process. We copy the data needed for each compute region onto the device (GPU) just before invoking that compute kernel. We define a data region that spans several compute regions to prevent frequent data transfers between the host (CPU) and the device (GPU).

### 2.2 Monte Carlo updates

Several Monte Carlo moves are attempted in REMC steps in I-TASSER. These updates/moves are accepted or rejected based on the Metropolis criterion (Metropolis *et al.*, 1953). Each of these moves is executed in parallel on the GPU by invoking parallel and kernel regions for the compute portions of the moves, where the data regions are spread over several compute areas. This optimizes the data region and reduces the time for communication.

### 2.3 Replica exchange

In REMC in I-TASSER, the replica exchange process happens after a set of Monte Carlo updates. The swaps by themselves do not require significant time. However, when multiple replicas are simulated and each swap requests for energy comparison of the involved replica pairs, the computational time becomes significant. Based on the approach of Gross *et al.* (2011), to lower the computational time for replica exchange, each replica is allocated in its own compute region involving multiple thread blocks, and updates are made available to the host from each replica. The Metropolis criterion is then used to access whether the exchange/swap should be accepted or not.

## 3 Results

We performed a comparative analysis of GPU-I-TASSER and CPU-I-TASSER (original I-TASSER) using a benchmark dataset of 71 proteins (Supplementary Table S1). After targeting the hardware of the GPU (in this case, Tesla P100 PCIe GPU), we obtained an average speedup of 10.27 $\times$  compared with the CPU (Intel Xeon E5-2680v3 processor) (Fig. 1). By comparing the predicted models with the native structure (results shown in Supplementary Table S2), we can see that GPU-I-TASSER predicts structures with comparable accuracy to I-TASSER in terms of TM-score (Zhang and Skolnick, 2004). We anticipate that GPU-I-TASSER will be useful in large-scale protein structure modeling applications in which the speed of the pipeline is essential. The complete source code is available for use from <https://zhanggroup.org/GPU-I-TASSER/>.

## Acknowledgement

The authors thank Mathew Colgrove of Nvidia for his tremendous help on this project and Dr Sunita Chandrasekaran for discussion.

## Funding

This work was supported in part by the National Science Foundation [DBI1564606 to D.K., IIS1901191, DBI2030790, MTM2025426 to Y.Z.]; the National Institute of General Medical Sciences [GM136422, S10OD026825 to Y.Z.] and the National Institute of Allergy and Infectious Diseases [AI134678 to Y.Z.].

*Conflict of Interest:* none declared.

## References

- Baek, M. *et al.* (2021) Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, **373**, 871–876.
- Gross, J. *et al.* (2011) Massively parallelized replica-exchange simulations of polymers on GPUs. *Comput. Phys. Commun.*, **182**, 1638–1644.
- Jumper, J. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.
- Kallberg, M. *et al.* (2014) RaptorX server: a resource for template-based protein structure modeling. *Methods Mol. Biol.*, **1137**, 17–27.
- Kryshtafovych, A. *et al.* (2019) Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins*, **87**, 1011–1020.
- Kryshtafovych, A. *et al.* (2021) Critical assessment of methods of protein structure prediction (CASP)-Round XIV. *Proteins*, **89**, 1607–1617.
- Metropolis, N. *et al.* (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087.
- Rohl, C.A. *et al.* (2004) Protein structure prediction using Rosetta. *Methods Enzymol.*, **383**, 66–93.
- Xu, D. and Zhang, Y. (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins*, **80**, 1715–1735.
- Yang, J. *et al.* (2015) The I-TASSER Suite: protein structure and function prediction. *Nat. Methods*, **12**, 7–8.
- Zhang, Y. and Skolnick, J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.
- Zheng, W. *et al.* (2019) LOMETS2: improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. *Nucleic Acids Res.*, **47**, W429–W436.