On Parameterizing Models of Antigen-Antibody Binding Dynamics on Surfaces – A Genetic Algorithm Approach and the Need for Speed

Daniel J. Burns[1] and Kevin T. May[2]

[1] Air Force Research Laboratory, Information Directorate, Rome, NY 13440 USA
burnds@rl.af.mil
[2] Department of Computer Engineering, Clarkson University, Potsdam, NY, USA 47907
mayk@clarkson.edu

Abstract. This paper discusses the performance of a simple GA for parameterizing a particular biomodel consisting of a set of coupled non-linear ordinary differential equations. Comments are offered on the need for speed that motivates choice of language and processing platform for solving scaled problems.

1 Introduction

Ag-Ab (Antigen-Antibody) binding dynamics at surfaces is of interest to biologists because of the critical need for biosensors and diagnostic tests for the presence of targeted substances in clinical, biological, or environmental samples [Zheng 1]. Accurate models of binding dynamics are needed to support the design and performance optimization of biosensor systems. Developing accurate models requires parameterizing them to fit experimental data. This is a hard optimization problem that can easily become analytically intractable for complex nonlinear bio-models. This motivates reduced order modeling that involves techniques such as variable replacement by exogenous functions, temporal windowing, sequential parameter fitting, etc. [Rundell 2]. Even with reduced order modeling, the computing time required for parameterizations can be many minutes. The present work evaluated whether a simple GA approach run on a full, unreduced model could be more efficient. This work served as a test case for developing working C codes that could be ported to parallel and embedded computer platforms to achieve extreme speed-ups known to be needed for certain hard problems.

2 Summary of the Work and Results

We wrote codes for simulating the Model and for parameterizing the Model with a simple GA in Labview and in MatLab, and translated them to C to achieve reasonable run times (100-10,000 X faster). Our GA used binary representations of the floating point Model parameters. Values were searched over adjustable width ranges that were different for each parameter. Adjustable GA parameters included # bits in each gene,

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 497–498, 2004.

chromosomes, # genes in each chromosome, # in initial population, # in running population, % of population selected to produce children, % of gene bits mutated at each generation, maximum # generations, ranking method for selection (probability based on fitness or rank), and termination criteria (based on total fitting error, or maximum single point error). We used uniform probability crossover for all genes, and elitism (keeping the current best individual).

The Model equations predict the# of Ag particles bound to an Ab functionalized surface, with one equation describing each attachment *valency* (i.e. the # of epitope sites binding an Ag particle). The Model has 6 parameters (initial association rate, initial dissociation rate, transport coefficient due to diffusion, transport coefficient due to gravity, crosslinking association rate, and crosslinking dissociation rate). A *complete* raw data set from the Model consisted of about 7 concentration vs time curves tracing out a binding and release experiment over 200 time points. Experimentally it is only possible to measure the *total* # of bound Ag, not the *valency* of a bound Ag. Therefore, we summed the curves to produce a single *total bound* concentration vs time curve. We measured the performance of the GA parameterization tool working with both *complete* data sets and reduced *total bound* data sets. The fitness function calculated the sum of the least squares differences of data sets from a *known* individual (with a set of preselected parameter values) and individuals in the population. Various supervisory programs were written to gather statistical performance results over typically 20 or 30 fitting runs, and to display results. Also, a number of metrics were defined to characterize the performance of the GA fitter (speed, accuracy, convergence) using different sets of GA parameters, and for determining how the metrics behaved with scaling, e.g. as a function of population sizes, maximum generations, etc.

We observed that fitting all six parameters to about 2% accuracy using *complete* data sets was easy using populations of about 200 run for 1000 generations. This involved about 200,000 function evaluations and took about 30 seconds. Times for fitting 5 parameters (without kg which hardly effects the data) were much better, typically about 2 seconds. We also determined that varying the ranges over which parameter values were fit (from 2 to 4 to 10 times the parameter values) changed the average number of generations required to converge from 500 to 800 to 1250.

Fitting all 6 Model parameters simultaneously from *total bound* data sets proved to be much harder, with typically only 10% of runs achieving <10% fitting errors. We are currently experimenting with a progressive GA (PGA) parameter fitting strategy that mimics the progressive fitting strategy described in [2] to improve competence.

References

1. Zheng, Y.; Rundell, A., “Biosensor Immuno-surface Engineering Inspired by B-cell Membrane Bound Antibodies: Modeling and Analysis of Multivalent Antigen Capture by Immobilized Antibodies, IEEE Transactions on NanoBioscience, 2(1):14-25, 2003.
2. Rundell, A.; DeCarlo, R.; Doerschuk, P.; HogenEsch, H.; “Parameter Identification for an Autonomous 11th Order Nonlinear Model of a Physiological Process”, Proceedings of the 1998 American Control Conference, 6: 3585-3589, 1998.