

NuMHC - Simulação do MHC

Helio Tsutomu Matayoshi NUSP: 5382308

Orientador: Prof. Marcelo Finger

Colaborador: Prof. Eduardo Finger

30 de novembro de 2008

Sumário

1	Introdução	2
2	Objetivo	2
3	Desafio	2
4	Conceitos e Tecnologias Estudadas	3
	4.1 Conceitos Biológicos	3
	4.2 Conceitos Técnicos	7
5	Atividades Realizadas	10
	5.1 Fase 1	10
	5.2 Fase 2	11
	5.3 Fase 3	12
6	Resultados e Produtos Obtidos	12
7	Conclusão	13

1 Introdução

O sistema imunológico, por inteiro, pode ser definido como um complexo emaranhado de reações químicas e que em muitas vezes não se é entendido completamente. Dentro deste sistema existe um elemento quem, se ele não existesse, o sistema como um todo não funcionaria. Ele é o MHC (*Major Histocompatibility Complex* — Complexo Maior de Histocompatibilidade). É nele que se encontram todas as informações sobre nossas próprias células e também as de agressores que nos atacaram em algum momento de nossas vidas. Com todas essas informações, ele é o responsável por coordenar todo o ataque aos organismos nocivos a nós.

2 Objetivo

Como o MHC é o coordenador de todo o sistema imunológico, é de se esperar que médicos tentem entender como ele funciona e, principalmente, como ele realiza a diferenciação entre agressores e células do próprio organismo. Pesquisas neste setor, envolvem vários experimentos que se utilizam de amostras do próprio corpo humano. Por isso, elas acabam sendo pesquisas longas (podendo durar meses) e, muitas vezes, também custosas.

Com o auxílio de um simulador do MHC, alguns pontos das pesquisas poderiam ser rapidamente resolvidos, implicando em utilizar amostragens reais com mais eficiência. Com esta idéia em mente, foi-se iniciado este projeto de criação de um simulador do MHC.

3 Desafio

Atualmente, existem alguns serviços que realizam algumas simulações, porém elas nem sempre podem ser consideradas corretas. Assim, foi-se considerado a idéia de criar um simulador através de modelagem de estados e verificações através de lógicas proposicionais. No entanto, poucas são as pesquisas realizadas neste ramo e, por isso, não há muitas referências que podem ser utilizadas, dificultando este trabalho e tornando-o bastante prolongado.

4 Conceitos e Tecnologias Estudadas

4.1 Conceitos Biológicos

Aminoácidos

Um aminoácido é uma molécula orgânica formada por átomos de carbono, hidrogênio, oxigênio, e nitrogênio unidos entre si de maneira característica.

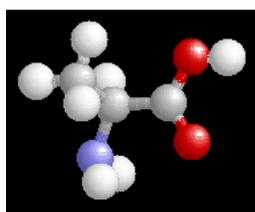
Alguns aminoácidos também podem conter enxofre. Os aminoácidos são divididos em quatro partes: o grupo amina (NH_2), grupo carboxílico (COOH), hidrogênio, carbono afa (todas partes se ligam a ele), e um radical característico de cada aminoácido.

Grupos

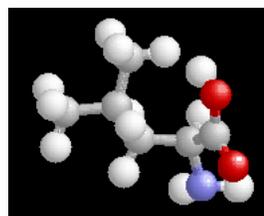
Os aminoácidos podem ser classificados de várias formas. No entanto, como forma de simplificar algo complexo, foi-se necessário seguir uma classificação que divide eles em quatro grupos: os hidrófobos, os que realizam ponte de hidrogênio, os aniônicos e os catiônicos.

- **Hidrófobos**

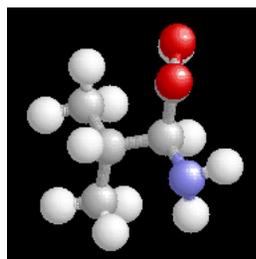
Estes aminoácidos tendem a fazer ligações com outros do mesmo grupo.



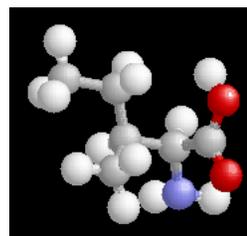
Alanina(a)



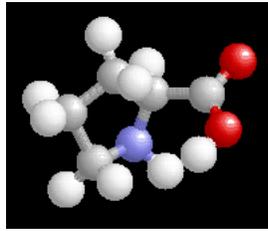
Leucina(l)



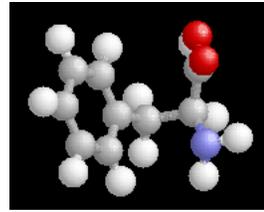
Valina(v)



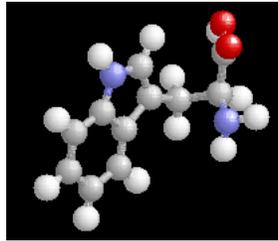
Isoleucina(i)



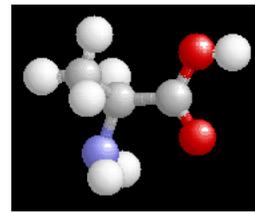
Prolina(p)



Fenilalanina(f)



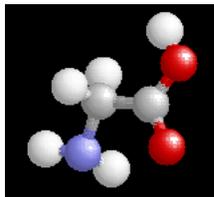
Triptofano(w)



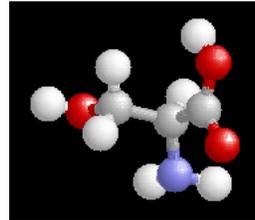
Metinina(m)

- **Ponte de Hidrogênio**

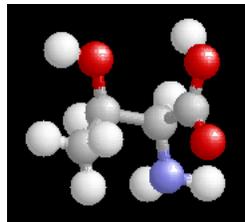
Estes aminoácidos tendem a fazer ligações com outros do mesmo grupo.



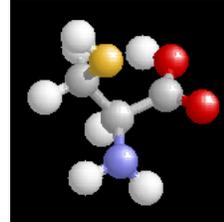
Glicina(g)



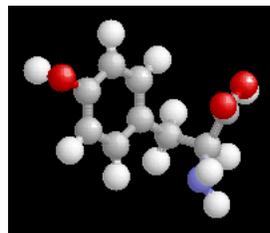
Serina(s)



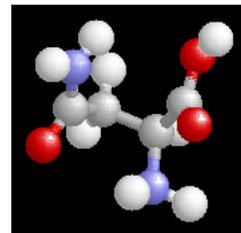
Treonina(t)



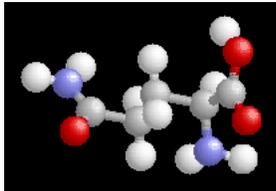
Cisteina(c)



Tirosina(y)



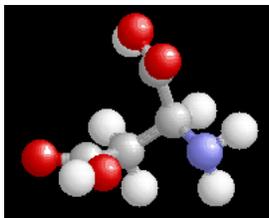
Asparagina(n)



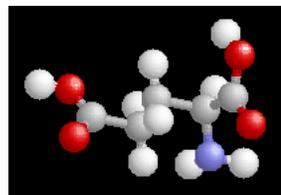
Glutamina(q)

- **Aniônicos**

Estes aminoácidos tendem a fazer ligações com outros do grupo dos catiônicos.



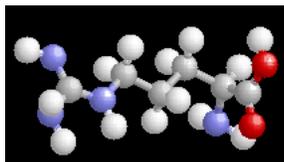
Aspartato(d)



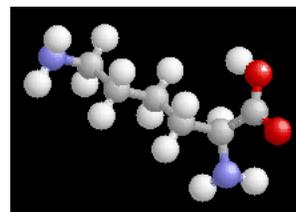
Glutamato(e)

- **Catiônicos**

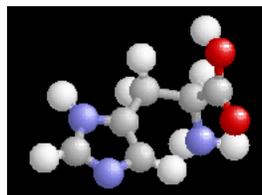
Estes aminoácidos tendem a fazer ligações com outros do grupo dos aniônicos.



Arginina(r)

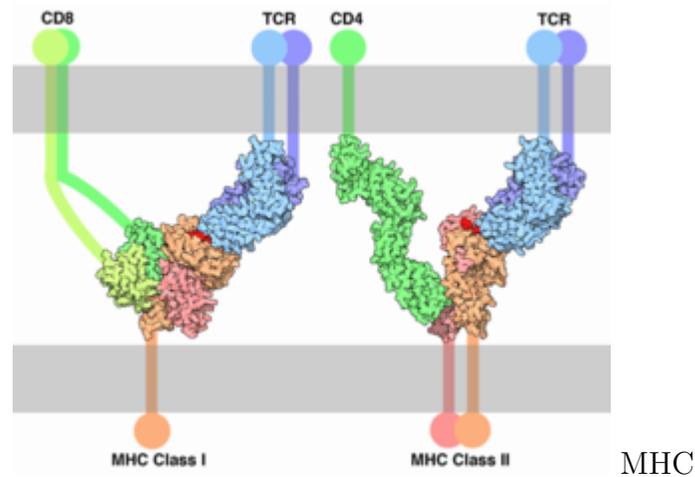


Lisina(k)



Histidina(h)

Major Histocompatibility Complex - Complexo Maior de Histocompatibilidade



O Complexo de Maior Histocompatibilidade (MHC) é uma grande região genômica ou uma família genes que, como dito anteriormente, possui uma importante tarefa no sistema imunológico. É nele que se encontram todas as informações sobre os antígenos do próprio ser humano (fragmentos de peptídeos das próprias células) e de antígenos externos (fragmentos de microorganismos invasores)[1] e são essas informações que ele apresenta para células brancas do sangue chamadas células T, que possuem a capacidade de matar ou coordenar o processo de extermínio de patógenos ou células infectadas ou defeituosas. [2]

Subgrupos

O MHC é dividido em três grupos[3] chamados MHC classe I, MHC classe II e MHC classe III:

- **MHC classe I** - Codifica peptídeos heterodiméricos de ligação de proteínas e também moléculas processadoras de antígenos.
- **MHC classe II** - Codifica peptídeos heterodiméricos de ligação de proteínas e proteínas que modulam antígenos carregados para as proteínas do MHC classe II pelos compartimentos lisossômicos como.
- **MHC classe III** - Codifica outros componentes imunes.

Reconhecimento

O MHC é formado por inúmeras proteínas que agem como indicadores de antígenos (tanto do indivíduo como de invasores externos) nas superfícies das células do indivíduo. Cada proteína é formada por uma cadeia de aminoácidos. Assim o MHC é capaz de reconhecer os antígenos através de fragmentos obtidos do elemento a ser identificado, ou seja, é verificado se os fragmentos (uma cadeia de aminoácidos) do elemento desconhecido faz ligação com alguma parte do MHC através das forças que interagem entre os aminoácidos. Caso tal ligação seja efetivada, então dependendo da região onde o fragmento se ligou o MHC o reconhece como algo próprio do hospedeiro ou como algo externo.

Caso se verifique que o antígeno visto seja do próprio hospedeiro e que nada se encontra fora do normal, então nenhuma precaução é tomada e todo o sistema imunológico é contido. No entanto, caso se verifique que o antígeno não pertença ao hospedeiro, o sistema imunológico verifica se ele pode ser um potencial agressor e, se necessário medidas urgentes são tomadas.

Há dois modos para que um antígeno externo seja processado[4] e reconhecido como tal:

- Através de células fagocíticas como os macrófagos, neutrófilos e monócitos que degradam partículas estrangeiras que são obtidas durante o processo de fagocitose. Logo a seguir, essas partículas são apresentadas para as moléculas do MHC classe II
- A outra maneira acontece quando uma célula do indivíduo é infectada por uma bactéria ou um vírus, ou quando ela vira uma célula cancerígena. Quando algum desses cenários ocorre, a célula tende a apresentar antígenos incomuns em sua superfície. Esses antígenos, não importando para que tipo de molécula do MHC que serão apresentadas, desencadeiam uma imunidade específica do corpo do hospedeiro.

4.2 Conceitos Técnicos

Modelo de estados

O modelo de estados é uma técnica de resolução de problemas complexos que consiste em estruturá-lo em vários estados, cada um com propri-

idades diferentes, e que estão interligados por fórmulas lógicas proposicionais.

A partir do modelo criado, é possível realizar a verificação de modelos, que é um processo onde, a partir de uma dada fórmula lógica, verificamos através de uma pesquisa exaustiva, se ela satisfaz o modelo.

NuSMV

NuSMV é um programa opensource que fornece uma linguagem para descrição de modelos de estados bem como a verificação deles através de expressões lógicas. O sistema SMV original foi escrito por K. McMillian em Carnegie Mellon University e, o NuSMV foi uma reimplementação dele, feita em Trento por A. Cimatti e M. Roveri.[5]

NuSMV permite checar sistemas de estados finitos através de especificações de lógica temporal CTL (Computation Tree Logic) [6]. A linguagem do NuSMV foi desenvolvida para permitir descrições de sistemas de estados finitos que podem ir desde modelos síncronos até modelos totalmente assíncronos. Ela também permite descrições hierárquicas e reutilizações de definições de componentes. O propósito básico da linguagem do NuSMV é a descrição, através de expressões de cálculo proposicional, de relações de transição entre estados finitos. Isso acaba fornecendo bastante flexibilidade na escrita do código, porém, ao mesmo tempo, traz também grandes perigos de inconsistência no código.

Programas em NuSMV consistem fortemente de um ou mais módulos, sendo que um deles precisa necessariamente ser chamado *main*. Módulos podem declarar variáveis e atribuir valores para elas. As atribuições vão desde valores iniciais até valores futuros que são determinados por expressões em termos dos valores atuais das variáveis. Essas expressões podem ser não-determinísticas, que são usadas para modelagem de ambientes e para abstrações.

Exemplo

O código a seguir é um pequeno exemplo[7] de entrada para o NuSMV:

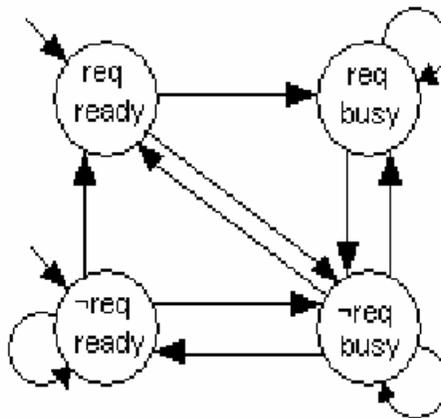
```

MODULE main
VAR
request : boolean;
status : {ready,busy}
ASSIGN
    init(status) := ready;
    next(status) := case
        request : busy;
        1 : {ready,busy};
    esac;

```

O programa contém duas variáveis: **request** de tipo **boolean**, onde 0 significa 'falso' e 1 'verdadeiro', e **status** do tipo enumeração, onse seu valor pode ser um dos dois {**ready**, **busy**}. Os valores iniciais e subsequentes da variável **request** não são determinados pelo programa, eles serão determinados aleatoriamente pelo próprio ambiente externo do modelo. Essa pseudo-especificação de **request** implica que o valor da variável **status** será determinada parcialmente: inicialmente, ela será **ready** e se tornará **busy** sempre que **request** for verdadeiro. Se **request** for falso, o próximo valor de **status** não é determinado.

A seguir, um esquema representando os estados criados pelo modelo escrito acima:



CTL

CTL (Computation Tree Logic)[7] é uma lógica do tipo branching-time, ou seja, seu modelo de tempo é uma estrutura do tipo árvore onde o futuro não é determinado; para cada modelo existe diferentes caminhos, onde qualquer um deles pode ser o caminho 'real' que é realizado.

Fórmulas CTL podem ser escritas da seguinte forma:

$$\phi ::= \perp | \top | p | (\neg\phi) | (\phi \wedge \phi) | (\phi \vee \phi) | (\phi \rightarrow \phi) | AX\phi | EX\phi | AF\phi | EF\phi \\ | AG\phi | EG\phi | A[\phi U \phi] | E[\phi U \phi]$$

Note que todos os conectivos temporais do CTL são formados por um par de símbolos. O primeiro dos símbolos é **A** ou **E**. **A** significa '*para todos os caminhos*' e **E**, '*existe pelo menos um caminho*'. O segundo símbolo pode ser **X**, **F**, **G** ou **U**, que significam, respectivamente, '*próximo estado*', '*algum futuro estado*', '*todos os futuros estados*' e '*até que*'. Estes pares de símbolos são indivisíveis, ou seja, não se pode escrever um deles sem o outro.

5 Atividades Realizadas

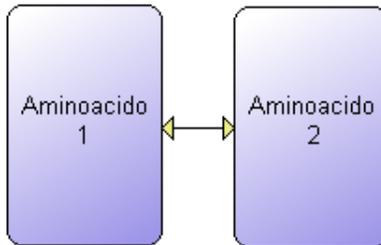
Inicialmente, o projeto foi dividido em três fases de planejamento:

5.1 Fase 1

Nesta fase, cada aminoácido foi modelado separadamente e, foi-se criado um modelo que verificava se um aminoácido se ligava com outro.

Para cada aminoácido, um módulo foi criado. Cada módulo contém as quatro propriedades consideradas para o projeto na forma de boolean, onde eles seriam 1 (verdadeiro) se o aminoácido contém a propriedade, ou 0 (false) caso não o contenha.

A verificação se ligação era feita nos próprios módulos dos aminoácidos. Foi-se passado para cada módulo o outro aminoácido testado e ele verificava se as propriedades dos dois eram compatíveis.

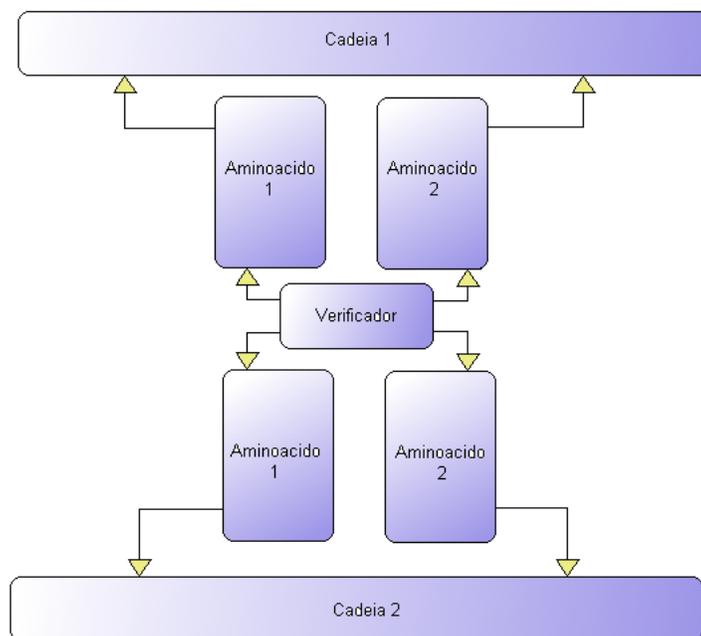


5.2 Fase 2

Nesta fase, cadeias simples, onde composições geométricas não foram levadas em consideração, foram criadas e, um modelo onde se verificava se duas cadeias de aminoácidos se ligavam.

A modelagem de cada cadeia se fez apenas em um único módulo. Cada módulo contém um vetor indicando os aminoácidos da cadeia. Cada cadeia foi representada através de dois apontadores que indicam duas posições diferentes dela.

Para verificar se as duas cadeias conseguem se ligar, um módulo diferente verifica se os dois aminoácidos apontados de cada cadeia se ligam fortemente e se eles estão a distâncias iguais.



5.3 Fase 3

Nesta fase, a modelagem foi praticamente mantida. Vetores de posicionamento foram adicionados para os módulos de aminoácidos e a verificação de ligação foi realizada através das posições.

Não foi identificada uma maneira eficaz de popular as posições automaticamente, por isso elas necessitam de preenchimento manual.

6 Resultados e Produtos Obtidos

No final deste projeto, foi-se obtido um programa, que mesmo incompleto, foi capaz de obter resultados satisfatórios para cadeias de tamanho pequeno, ou seja, o modelamento criado consegue simular com perfeição as ligações que podem ocorrer para cadeias deste tipo.

No entanto, para cadeias de tamanho maior, por elas formarem estruturas tridimensionais complexas, uma modelagem real delas se tornou bastante

custoso e trabalhoso, pois elas precisavam necessariamente ser modeladas manualmente, ou seja, todas as variáveis que representam as propriedades dos aminoácidos devem ser descritas pessoalmente. Também pode se dizer que por terem um tamanho grande, existem interações mais complexas entre os aminoácidos que não foram cogitados nesta implementação, tais como as interações entre os aminoácidos da própria cadeia.

7 Conclusão

Pode se concluir que um simulador do MHC pode ser construído através de modelos de estados e verificações lógicas e, mesmo o projeto tendo apresentado resultados insatisfatórios é possível ajustá-lo de forma a apresentar resultados melhores futuramente.

Referências Bibliográficas

- [1] Imgt/hla database, 2008, <http://www.ebi.ac.uk/imgt/hla/>.
- [2] C. A. Janeway, P. Travers, M. Walport, and M. J. Shlomchik, *Imunobiologia - O Sistema Imune na Saúde e na Doença*, 6 ed. (Artmed Editora, 2006).
- [3] Major histocompatibility complex, 2008, <http://www.cehs.siu.edu/fix/medmicro/mhc.htm>.
- [4] A. Aderem and D. Underhill, Annual Review of Immunology (1999).
- [5] R. Cavada *et al.*, *NuSMV 2.4 User Manual*.
- [6] Nusmv, 2008, <http://nusmv.itc.it/>.
- [7] M. Huth and M. Ryan, *Logic in Computer Science - Modelling and Reasoning about Systems*, 2 ed. (Cambridge, 2006).