

NuMHC - Simulação do MHC

Helio Tsutomu Matayoshi NUSP: 5382308

Orientador: Prof. Marcelo Finger

Colaborador: Prof. Eduardo Finger

7 de janeiro de 2009

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 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 que, se não existisse, 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 de agressores que nos atacaram em algum momento de nossas vidas. Com todas estas informações, ele é o responsável por coordenar 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, como por exemplo, seria possível descobrir se um determinado elemento seria reconhecido pelo MHC como um agressor ou não antes de utilizá-lo em um paciente. Isto implicaria em utilizar amostragens reais com mais eficiência. Com esta idéia em mente, começou-se 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, surgiu a idéia de criar um simulador através de modelagem de estados e verificações através de lógicas proposicionais. No entanto, existem poucas pesquisas realizadas neste ramo e, por isso, não há muitas referências que possam 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.

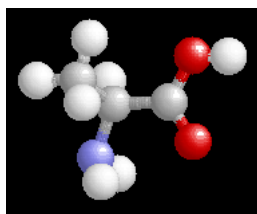
Os aminoácidos são divididos em quatro partes: o grupo amina (NH_2), grupo carboxílico (COOH), hidrogênio, carbono alfa (todas as partes se ligam a ele), e um radical característico de cada aminoácido.

Grupos

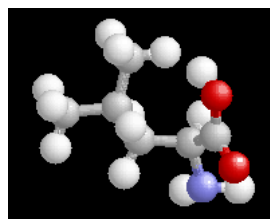
Os aminoácidos podem ser classificados de várias formas. No entanto, para simplificar a diferenciação entre eles e, assim, facilitar o desenvolvimento do projeto, a classificação dos aminoácidos adotada foi a que divide eles em quatro grupos distintos: os hidrófobos, os que realizam ponte de hidrogênio, os aniônicos e os catiônicos.

- **Hidrófobos**

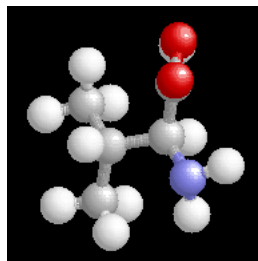
Estes aminoácidos tendem a fazer ligações com outros do mesmo grupo.



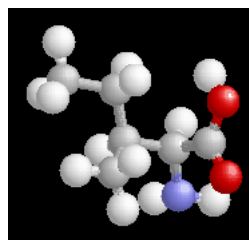
Alanina(a)



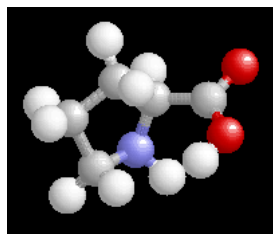
Leucina(l)



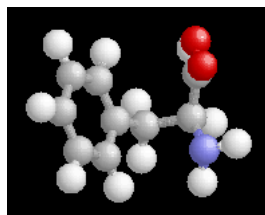
Valina(v)



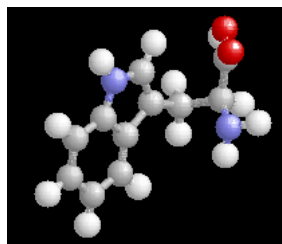
Isoleucina(i)



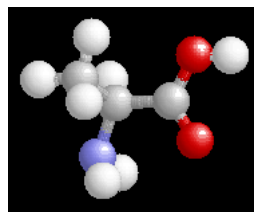
Prolina(p)



Fenilalanina(f)



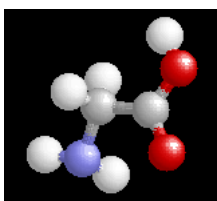
Triptofano(w)



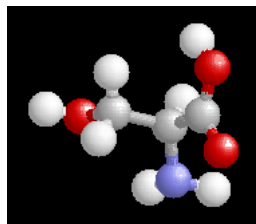
Metinina(m)

• Ponte de Hidrogênio

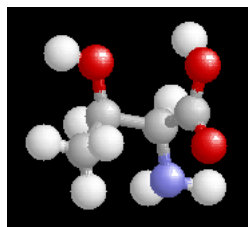
Estes aminoácidos tendem a fazer ligações com outros do mesmo grupo.



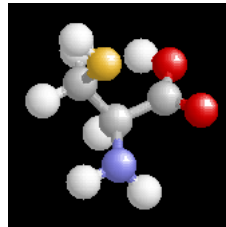
Glicina(g)



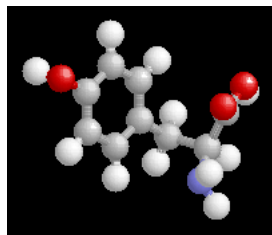
Serina(s)



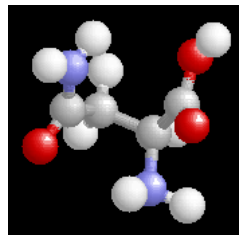
Treonina(t)



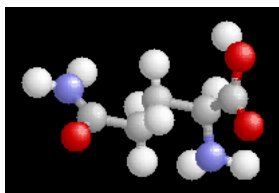
Cisteina(c)



Tirosina(y)



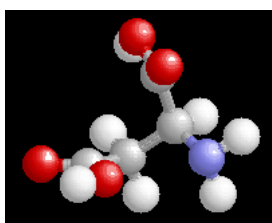
Asparagina(n)



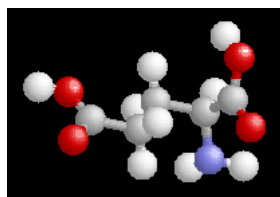
Glutamina(q)

- **Aniônicos**

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



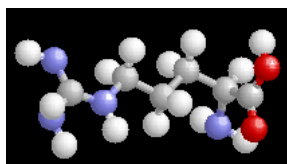
Aspartato(d)



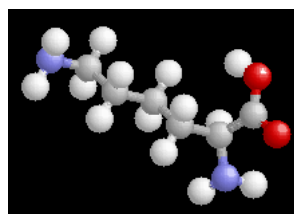
Glutamato(e)

- **Catiônicos**

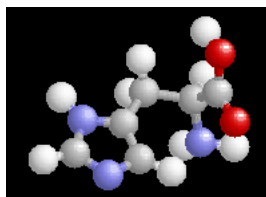
Estes aminoácidos tendem a fazer ligações com outros do grupo dos aniônicos.



Arginina(r)

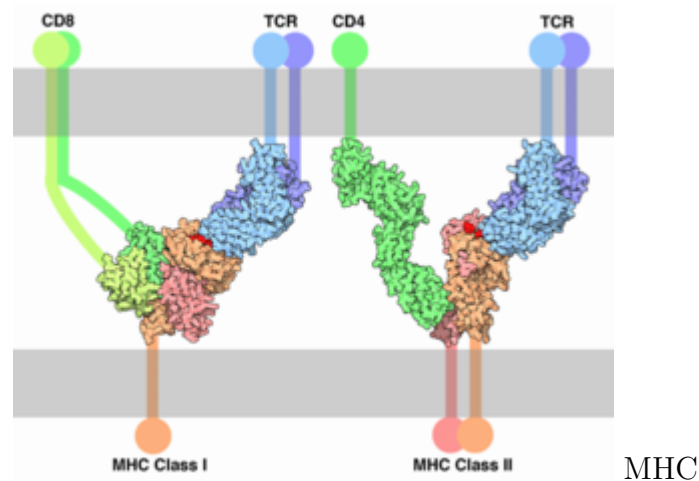


Lisina(k)



Histidina(h)

Major Histocompatibility Complex - Complexo Maior de Histocompatibilidade



O Complexo Maior de Histocompatibilidade (MHC) é uma grande região genômica ou uma família de 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 estas informações que ele apresenta para as células brancas do sangue, chamadas células T, que possuem a capacidade de matar ou codenar 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.
- **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. 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 realiza uma ligação com alguma parte do MHC através das forças que interagem entre os aminoácidos. Caso tal ligação seja efetivada, então dependendo da região onde esta ligação ocorreu, o MHC o reconhece como algo próprio do hospedeiro ou como algo externo.

Caso se verifique que o antígeno pesquisado 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 se torna uma célula cancerígena. Quando algum destes cenários ocorre, a célula tende a apresentar antígenos incomuns em sua superfície. Estes antígenos, não importando para que tipo de molécula do MHC sejam apresentadas, desencadearão uma imunidade específica no 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-los em vários estados, cada um com propriedades

diferentes, e que estão interligados por fórmulas lógicas proposicionais.

A partir do modelo criado, é possível verificar se uma dada fórmula lógica satisfaz o modelo, através de uma pesquisa exaustiva.

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. McMillan em Carnegie Mellon University e o NuSMV foi uma reimplementação dele, feita em Trento por A. Cimatti e M. Roveri.[5]

O NuSMV permite checar sistemas de estados finitos através de especificações de lógica temporal CTL (Computation Tree Logic) [6]. A linguagem do NuSMV foi desenvolvida afim de 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. Isto fornece bastante flexibilidade na escrita do código, porém, ao mesmo tempo, traz também grandes perigos de inconsistência no código.

Programas em NuSMV consistem de um ou mais módulos, sendo que um deles deverá ser chamado de *main*, para indicar ao NuSMV qual deles será o controlador geral. Os 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. Estas expressões podem ser não-determinísticas, normalmente utilizadas para a modelagem de ambientes e para abstrações.

Exemplo

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

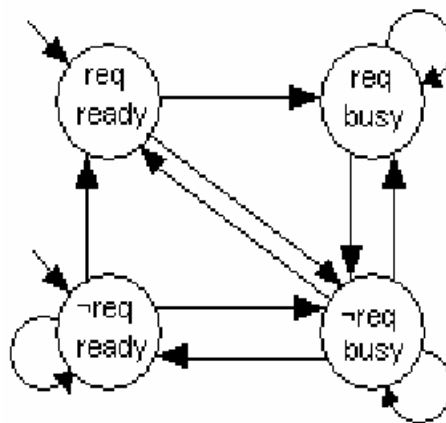
```

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** do tipo **boolean**, onde 0 significa 'falso' e 1 'verdadeiro', e **status** do tipo enumeração, onde seu valor pode ser um dos dois: **{ready, busy}**. Os valores iniciais e subseqüentes 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 existem diferentes caminhos, onde qualquer um deles pode ser o caminho 'real' a ser percorrido.

Fórmulas CTL podem ser escritas da seguinte forma:

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

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 estado futuro*', '*todos os estados futuros*' e '*até que*'. Estes pares de símbolos são indivisíveis, ou seja, não se pode escrever um deles sem o outro.

5 Atividades Realizadas

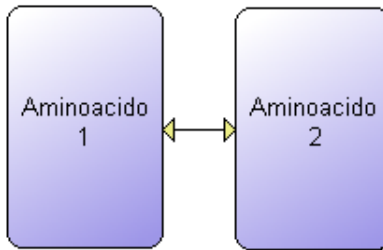
O projeto foi dividido em três fases de planejamento:

5.1 Fase 1

Nesta fase, cada aminoácido foi modelado separadamente e, as verificações de existência de ligações eram realizadas somente entre dois aminoácidos.

Para cada aminoácido, existia um módulo que os representava. Cada módulo continha as quatro propriedades dos aminoácidos (hidrófobo, ponte de hidrogênio, ânionicos e catiônicos), todas do tipo boolean, onde 1 (verdadeiro) significava que o aminoácido continha a propriedade, e 0 (falso) caso ela não lhe pertencia.

A verificação de existência de ligações era feita nos próprios módulos dos aminoácidos. Dois aminoácidos eram escolhidos, para cada um era passado o outro aminoácido e, ambos verificavam se as propriedades deles eram compatíveis um com o outro.

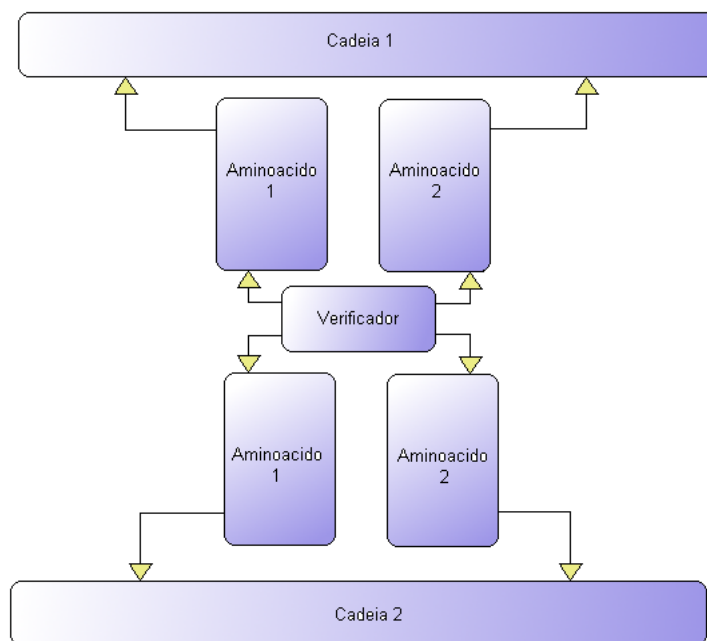


5.2 Fase 2

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

A modelagem de cada cadeia se fez apenas em um único módulo. Cada módulo continha um vetor indicando os aminoácidos da cadeia. Os aminoácidos de cada cadeia foram representados através de dois apontadores que indicavam duas posições diferentes.

Para verificar se as duas cadeias conseguiam se ligar, um outro módulo verificava se os dois aminoácidos apontados em cada cadeia se ligavam fortemente e se eles estavam a distâncias iguais (era verificado se a diferença dos índices dos vetores de cada cadeia eram iguais).



5.3 Fase 3

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

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

6 Resultados e Produtos Obtidos

Os testes realizados para verificar se o código criado gera modelos con-
dizentes com os fatos reais observados em laboratório consistiam em passar
duas cadeias de aminoácidos e verificar se o código era capaz de responder
corretamente se as cadeias se ligam ou não. Os testes também levavam em
consideração o tempo que o código necessitava para dar a sua resposta. Ini-
cialmente, foram testadas cadeias de dois aminoácidos e este número era
aumentado gradualmente.

Para cadeias de até sete aminoácidos, o código respondia com certa precisão aos testes (cerca de 90% deles eram respondidos corretamente) e suas repostas eram obtidas imediatamente após ser chamado. Para cadeias de tamanho maior do que sete aminoácidos, o código respondia corretamente às verificações, porém ele necessitava de mais tempo para responder, sendo que quanto maior era o tamanho da cadeia, maior era o tempo gasto.

O tempo necessário para verificações de cadeias de até dez aminoácidos era por volta de alguns minutos. No entanto, este tempo aumentava rapidamente com o aumento do tamanho da cadeia e, para cadeias de quinze aminoácidos, o tempo necessário era praticamente de um dia. Com o aumento do tamanho das cadeias, o código começou a demonstrar pequenas quedas em sua taxa de acertos, para cadeias de dez aminoácidos ele acertava 82% dos testes. Esta diminuição na taxa de acertos é percebida pois quanto maior for o tamanho das cadeias, mais complexas suas estruturas tridimensionais se tornam e também há a possibilidade de influências de reações não previstas pela implementação, como as interações entre os aminoácidos da própria cadeia.

7 Conclusão

Cadeias de aminoácidos reais podem possuir mais do que trinta aminoácidos. Apesar do código criado neste projeto responder corretamente à maioria dos testes, o tempo necessário para que o código envie a resposta de verificação para cadeias deste tamanho é grande demais para competir com verificadores já existentes. Logo, o código final criado por este projeto se torna inviável para que médicos o utilizem como ferramenta para suas pesquisas.

No entanto, há diversas melhorias que podem ser realizadas. Como o código responde bem para mais da metade dos testes realizados, é necessário otimizá-lo para que suas respostas sejam retornadas em tempo competitivo com os verificadores atuais. Também há a necessidade de automatizar a modelagem das estruturas tridimensionais, o que diminuirá o trabalho realizado pelo modelador e também diminuirá possíveis erros de modelagem. Uma outra melhoria que poderia ser feita é a pesquisa de diferentes iterações entre os aminoácidos, para tornar as simulações mais parecidas com as observações

reais.

A partir destes resultados, é possível verificar que um simulador do MHC pode ser construído através de modelos de estados e verificações lógicas, porém ainda há melhorias que necessitam serem feitas antes que este programa possa ser considerado como uma ferramenta para pesquisas médicas.

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).