

Copyright (c) Vladimir DARIC

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

# Recherche de motifs dans une séquence ADN au moyen d'un automate

## Introduction

Il s'agit ici d'implémenter un automate capable de reconnaître un motif dans une séquence. Ce dernier est construit de telle sorte qu'à chaque occurrence de  $x$  dans le texte, il passe par un état particulier. La lecture du texte s'effectue en temps linéaire. L'automate lui-même est construit à l'avance. Construction de celui-ci par un algorithme peut s'avérer fastidieuse. L'algorithme de *Knuth-Morris-Pratt* peut être une alternative permettant de contourner ce problème, mais ceci n'est pas le sujet de notre travail.

L'utilisation d'un tel automate pour la recherche d'un motif simple (pas de répétitions internes) de type « ATG » n'est pas une meilleure solution que d'utiliser l'algorithme naïf. Le nombre de comparaisons effectués est exactement le même. L'utilisation d'un automate prend tout son sens lorsqu'on cherche un motif ayant lui-même des répétitions de suffixes (« **AGCTAGCCA** »). De plus il est extrêmement facile de rechercher une expression régulière dans un texte à l'aide d'un automate. En effet, le temps de recherche d'un motif de longueur  $n$  est le même que celui nécessaire pour trouver une expression régulière de même longueur.

Les schémas des automates ont été joints en annexe.

## Initialisation du programme

Le programme est lancé par la commande :

```
$> java Chercher fic_automate fic_sequence
```

Il est possible de lancer le programme avec l'option « -h ». Une aide utilisateur est alors proposée.

Seulement 2 arguments sont acceptés, une autre manière d'initialiser le programme provoque un message d'erreur et l'arrêt de celui-ci.

## JAVADOC

Package substringFinder

[Class Summary](#)

[Class](#)

[Automate](#)

[CharSwr](#)

[Chercher](#)

[Etat](#)

Class Automate

**substringFinder.Automate**

#### Constructor Summary

**Automate**( *String* fic\_automate) - *fic\_automate* contient une suite de nombres

#### Method Summary

<i>Etat</i>	<b>getEtatAt</b> ( <i>int</i> position) - renvoie l' Etat de trouvant à une position donné
<i>int</i>	<b>NbEtats</b> () - méthode renvoie le nombre total d'états
<i>void</i>	<b>printAutomate</b> ()

#### Constructor Detail

Automate

*public Automate*(*java.lang.String* fic\_automate)

*throws java.io.FileNotFoundException,*

*java.io.IOException*

#### Method Detail

nbEtats

*public int nbEtats*()

*methode renvoie le nombre total d'etats*

getEtatAt

*public Etat getEtatAt*(*int* position)

printAutomate

*public void printAutomate*()

## Class Etat

## Constructor Summary

<code><a href="#">Etat</a>(int a, int b, int c, int d)</code> – assigne pour chaque base (ordre G, C, T, A) l'état suivant
--

## Method Summary

int	<code><a href="#">getPosEtat</a>()</code> – renvoie la place de l'Etat courant dans le vector
int	<code><a href="#">nextEtat</a>(int nucleotide)</code> – renvoie le numéro de l'Etat vers lequel l'Automate doit transiter
void	<code><a href="#">printEtat</a>()</code>

## Constructor Detail

Etat

```
public Etat(int a, int b, int c, int d)
```

## Method Detail

getPosEtat

```
public int getPosEtat()
```

nextEtat

```
public int nextEtat(int nucleotide)
```

printEtat

```
public void printEtat()
```

## Class CharSwr

## Constructor Summary

<a href="#">CharSwr</a> ( <i>java.lang.String le_fic</i> )	
--	--

## Method Summary

<i>int</i>	<a href="#">getPosition</a> () - <i>donne la position de lecture</i>
<i>boolean</i>	<a href="#">isOpen</a> () - <i>verifie si le fichier est ouvert/accesible</i>
<i>int</i>	<a href="#">nextChar</a> () - <i>effectue une lecture. Les caractères autres que a, c, t, g, A, C, T et G sont ignorés</i>
<i>int</i>	<a href="#">nextNum</a> () - <i>effectue une lecture</i>

## Constructor Detail

## CharSwr

```
public CharSwr(java.lang.String le_fic)
    throws java.io.FileNotFoundException
```

## Method Detail

## nextNum

```
public int nextNum()
    throws java.io.FileNotFoundException,
           java.io.IOException
```

## Throws:

```
java.io.FileNotFoundException
java.io.IOException
```

## nextChar

```
public int nextChar()
    throws java.io.FileNotFoundException,
           java.io.IOException
```

## Throws:

```
java.io.FileNotFoundException
java.io.IOException
```

## getPosition

```
public int getPosition()
```

## isOpen

```
public boolean isOpen()
    throws java.io.IOException
```

## Throws:

```
java.io.IOException
```

**Annexe 1 : Exemple d'exécution :**

```
$ > java expressionReg.automate vezatin.sequence
OK on y va
Motif trouve a la position 175
Motif trouve a la position 245
Motif trouve a la position 572
Motif trouve a la position 723
Motif trouve a la position 783
Motif trouve a la position 853
Motif trouve a la position 1150
Motif trouve a la position 1643
Motif trouve a la position 1718
Motif trouve a la position 1772
Motif trouve a la position 2012
Motif trouve a la position 2139
Motif trouve a la position 2287
Motif trouve a la position 2362
Motif trouve a la position 2461
Motif trouve a la position 2629
Motif trouve a la position 2672
Motif trouve a la position 2824
Motif trouve a la position 2894
Motif trouve a la position 2949
Motif trouve a la position 3010
Motif trouve a la position 3064
Motif trouve a la position 3140
Motif trouve a la position 3164
Motif trouve a la position 3208
Motif trouve a la position 3382
Motif trouve a la position 3430
Motif trouve a la position 3825
Motif trouve a la position 3851
Motif trouve a la position 3892
Motif trouve a la position 3929
Motif trouve a la position 4227
Motif trouve a la position 4305
Motif trouve a la position 4342
Motif trouve a la position 4386
Motif trouve a la position 4458
Motif trouve a la position 4491
Motif trouve a la position 4531
Motif trouve a la position 4598
Motif trouve a la position 4815
Motif trouve a la position 4865
Motif trouve a la position 4885
```

```
Le motif a ete trouve 42 fois.
```

**Annexe 2 : Le fichier vezatin.sequence :**

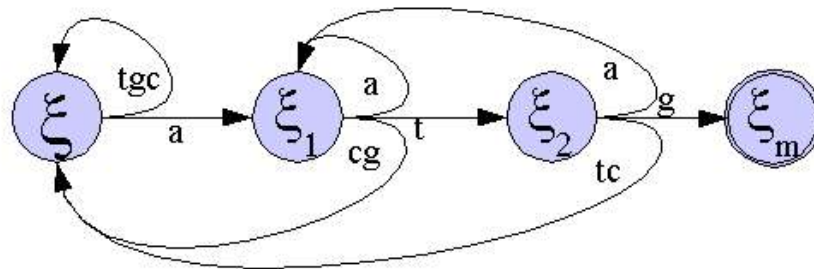
```

1 tccactttac caatacttac aggatctggg acacacagac tttgaaatat gttcttcttt
61 gtcacaaaa acagaaaaat gcacaacaga gggacaacaa aagcctccta caagagtcct
121 accaaaatac ctgggatata gtaatcactc aatgaatata aactgcactt actggcatgc
181 tcaaggaatg ggctattaag caaggatatcc tgttaaaagt ggctgaaacc atcaaaagtt
241 ggattttttt ttctcagtgc aataagaaag atgacttact tcacaagttg gatattggat
301 tccgactcga ctcatcatc accatcctgc aacaggaagt cctgttacia gaggatgtgg
361 agctgattga gctacttgat cccagtatcc tgtctgcagg gcaatctcaa caacaggaaa
421 atggacacct tccaacactt tgctccctgg caaccoctaa tatttgggat ctctcaatgc
481 tatttgccct cattagcttg ctcgttatgc ttcccacttg gtggattgtg tcttccaggc
541 tggtatgggg agtgattcta tttgtgtatc tggtcataag agctttgaga ctatggagga
601 cagccaaact acaagtgacc ctaaaaaaat acagcgttca tttggaagat atggccacaa
661 acagccgagc ttttactaac ctctgtgaga aagctttacg tctcattcaa gaaccggaag
721 ttgattccagc aggatttaca ctggtcagtg ctgcttgccc atttaataaa gctggacagc
781 atccaagtca gcattctcatc ggtccttogg aagctgtcta cogaactcta agagccaact
841 tccaagcagc aaggctagct accctatata tgctgaaaaa ctacccctcg aactctgaga
901 gtgacaatgt aaccaactac atctgtgtgg tgccttttaa agagctgggc ctggactta
961 gtgaagagca gatttcagaa gaggaagcac ataactttac agatggcttc agcctgcctg
1021 cattgaaggt ttgtttccaa ctctgggtgg cacagagttc agagtcttc agacggttag
1081 ccttattact ttctacagcc aattcaacct ctgggcoctt acttactcca gcaactctgc
1141 ctcatcgtat ctatctgat gtgactcaag gtctacctca tgctcattct gcctgtttgg
1201 aagagcttaa gcgcagctat gagtctatc ggtactttga aactcagcac cagtcagtac
1261 cgcagtgttt atccaaaact caacagaagt caagagaact gaataatgtt cacacagcag
1321 tgcgtagctt gcagctccat ctgaaagcat tactgaatga ggaataaatt cttgaagatg
1381 aacttgaaaa gcttgtttgt actaaagaaa cacaagaact agtgtcagag gcttattcca
1441 tcctagaaca gaaattaaag ttgattcagc cccacgttca agcaagcaac aattgtggg
1501 aagaggccat ttctcagatc gacaaactgc tacgaagaaa tacagataaa aaaggcaagc
1561 ctgaaatagc atgtgaaaaa ccacattgta cagtagtacc tttgaagcag cctactctac
1621 acattgcaga caaagatcca atcccagag agcaggaatt agaagcttat gtatagata
1681 tagatattga taagtatttc agaaaggatg atttttatta cttgtctcaa gaagacaag
1741 agagacagaa gcgtgagcat gaagaatcca agagggtgct ccaagaatta aaatctgtgc
1801 tgggatttaa agcttcagag gcagaaaggc agaagtggaa gcaacttcta ttagtgatc
1861 atggggtaaa gtccgcatgg aattagaggg ggaagtctg cactggaacc tgctgtggcc
1921 attttttttc tgttgttgct gcttctctag tggagttact gcagcttagg gatggttccct
1981 ggaagtggac acattttaca aaaaactttt taaatacac gccagctctt cgagaagcac
2041 ttttttctct ccgtcatgct tcatactata ttgtattgta ctaaccctc atgctgggat
2101 gagcgactgc ctgcaagca accactgcc tactaacatc aagaggggct aaataaact
2161 gcattttctc tgcaagggac aacttctacc gtatgaatat atattatatt tatgtattat
2221 attttgatac atagtgtatc ttgaatgcag acttcatctg tattttctgt ggtactcaa
2281 aatgtgtaga agttatggct tttctttcat tagggctctg ttagggtggtt ggaagtaag
2341 ttaaattgta tgttgcatct ttttccatgc tgttttaact tgtaaatata tatgtataca
2401 tgtggaagat attattatta tctagattac ttaagcctt ttatctgatg tattatcagt
2461 gaaaaccatg ctgcttattc cctaatacaa tggttctcaa acttttagtgt gtatcagaat
2521 caagatgagg cctgttaaaa tatagctggt cccaactccc agagtttcca ttcagaggg
2581 tctgggatca ggccaagaa tttgcatctt taacaaatc ctagggtgata actgataatg
2641 ttggttcagg gaacacactt tcagaaccac tgcttataa ttttaataa gataatattg
2701 gaggtgtcat aaaaaccttc atcaaatccc tatctcattt catttcttag gcattttctc
2761 tttataaaag tttcctggat ttataaaatt tcattgattt tactagaaag ttttatttag
2821 cttttattta tttattgttc ttttaagaa aaaaactgag agtttgaatt aaaagtttaa
2881 atcattgcta ggtccttagt tatttttttt ctttttctt tttttttatt tgaagccgtg
2941 ttgaaatcct tgcctcctgt agaccagtg gaaccataa gtaattcaga accatcaatg
3001 aatcagata tgggaaaagt cagtaaaaat gatactgaag aggaaagtaa taatccgcca
3061 caacagaaa tgaataaagt aggactgagt atttatgtga aaactctcta gaaggtaaaa
3121 ataaagataa ttcttcaaat gaagtcttcc ccaaggagc agaagaaaga atgtgttacc
3181 aatgtgagag tgaagatgaa ccacaagcag atggaagtgg tctgaccact gccctccaa
3241 ctcccaggga ctcatcagc ccctccatta agcagaggct ggcacggcta cagctgtcac
3301 cagattttac ctctactgct ggctgtgctg cagaagtggc tgctagatct ctctccttta
3361 ccaccatgat ggaacagact tttgggtgat aggaggaaga acaataata gaagaaaaa
3421 aaaatgagat agaagaaaag taagaaccaa gattcatatg aagtgatatt agattgttcc
3481 ttttcaaaaa gtgtttagct tcaagactgg aaagggaata tgagtgtgaa tttactatat
3541 gtaaagctaa gatgtggatt tacaggaaga accctggttt gaataactga tctgaaatta
3601 gtagttacct gtaaattgca gatcttttag gaaaataaga gaaaggtaag ggctcttttg
3661 aataaactgc tgttttattt gtggcacaac tgatcaatct tggaaattct ttaagtattt
3721 ttaataagaa atgaattatc atttcttgcc agaatttgct accttaaggt gattgggaaa
3781 attctgttgc aagaacatta acatttagta tgactccttt ttactgtatt cttgcagtta
3841 ataactgcag ctattatggt aataacaagt tgtttgtatt ttatttttgt ttataaccagt
3901 cttaaagatc caggttctga ataaaaaaat taattgatac aaaaaaaaaa gctagcca

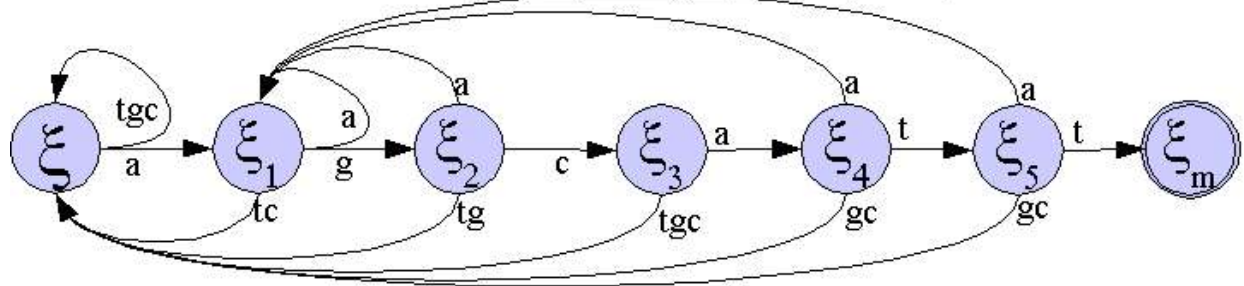
```

**Annexe 3 : Les schémas des automates**

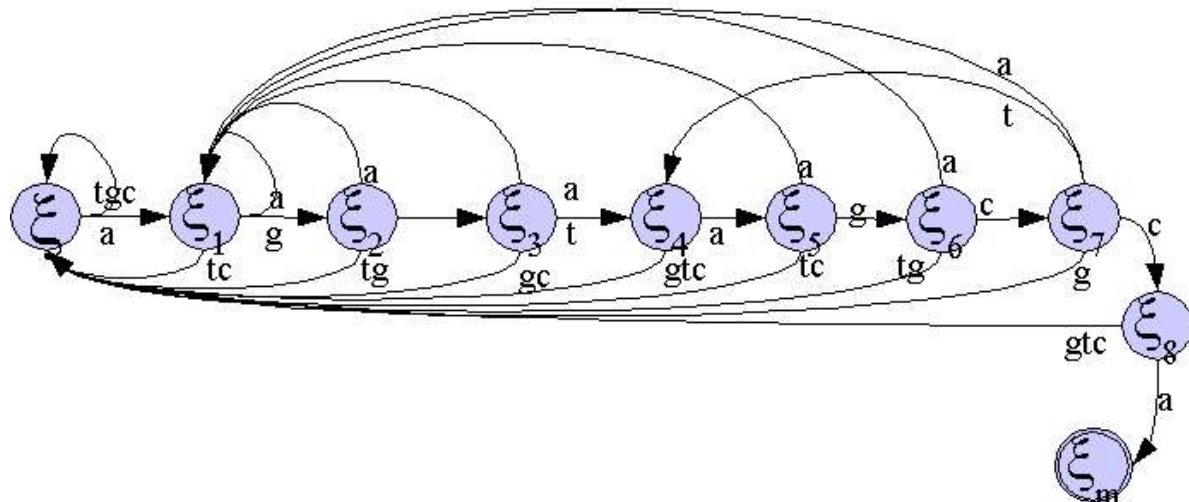
**Automate ATG - 0001|0021|3001**



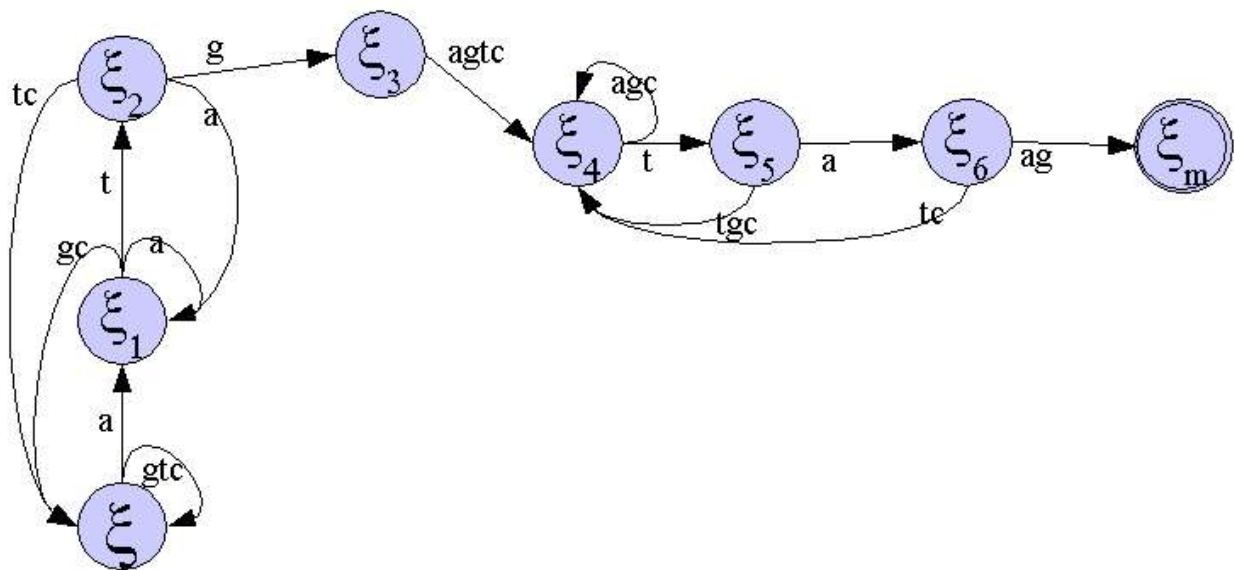
**Automate AGCATT - 0001|2001|0301|0004|0051|0061**



**Automate AGCTAGCCA - 0001|2001|0301|0041|0005|6001|0701|0841|0009**



# Automate $ATG(A|T|G|C)^*TA(A|G)$ - 0001|0021|3001|4444|4454|4446|7447



## Annexe 4 : Le code

```

/**
 * La classe Chercher construit l'automate demande
 * puis le lance. Elle lit également, de maniere sequentielle
 * le fichier contenant la sequence et lance la boucle de comparaisons.
 * Accepte en entree les deux noms de fichiers, l'option "-h" donne
 * une aide utilisateur
 * @see Automate
 * @see Etat
 * @see CharSwr
 */

import java.io.FileNotFoundException;
import java.io.IOException;

public class Chercher {
public static void main(String[] args) throws FileNotFoundException , IOException {

switch (args.length) {
    case 0 : case 1 :
        if ( args[0].compareTo("-h") == 0 ) {
            System.out.println("Usage: >>java Chercher fic_motif fic_sequence");
        } else {
            System.out.println("ERREUR : Trop peu de parametres");
            System.out.println("option -h pour obtenir une aide utilisateur");
        }
        break;
    case 2 :
        System.out.println("OK on y va");

        ///////////////////////////////////////////////////////////////////
        /////////////// ici commencent les choses serieuses
        ///////////////////////////////////////////////////////////////////
    try {
        Automate r2d2 = new Automate (args[0]);
        CharSwr sequence = new CharSwr(args[1]);
        Etat etat_courant = r2d2.getEtatAt(0);
        int no_etat_courant = etat_courant.getPosEtat(); //qui est apriori 0
        int tmp;
        int nb_etats;
        int nb_motifs=0;

        while ( sequence.isOpen() ) {
            tmp = sequence.nextChar();
            if (tmp !=0) {
                //System.out.println("num etat courant : " + no_etat_courant);
                no_etat_courant = etat_courant.nextEtat(tmp);

                if (no_etat_courant == r2d2.nbEtats() ) {
                    int pos = sequence.getPosition() - no_etat_courant ;
                    //System.out.println("no etat courant : " + no_etat_courant);
                    System.out.println("Motif trouve a la position " + pos );
                    nb_motifs++;
                    //et on relance l'automate
                    etat_courant = r2d2.getEtatAt(0);
                } else {
                    etat_courant = r2d2.getEtatAt(no_etat_courant);
                }
            }
        }

        System.out.println();
        System.out.println("Le motif a ete trouve " + nb_motifs + " fois.");

    } catch (FileNotFoundException e) {
        System.err.println("Fichier inexistant : " + e);
        throw(e);
    } catch (IOException e) {
        System.err.println("IOError:" + e);
        throw(e);
    }
    break;
    ///////////////////////////////////////////////////////////////////
    /////////////// point de non retour
    ///////////////////////////////////////////////////////////////////
    default :
        System.out.println("ERREUR : trop de parametres");
        System.out.println("option -h pour obtenir une aide utilisateur");
        break;
    }
}
}
}

```

```

/**
 * La classe automate construit un collection des etats
 * et les stoque dans un vecteur
 * Le dernier etat est considere comme ETAT FINAL
 */

import java.io.*;
import java.util.Vector ;
import java.io.FileNotFoundException;
import java.io.IOException;

public class Automate {

private int base_1 , base_2 , base_3 , base_4 ;
private CharSwr fic_automate;
private Etat state;
private Vector states = new Vector();
private CharSwr desc_fic ;
/*
 * constructeur de la classe Automate, construit les automates
 * en allant piocher les fic *.sequence
 * @param fic_automate le nom du fichier contenant la structure de l'automate
 */

public Automate ( String fic_automate ) throws FileNotFoundException, IOException {
    try {
        desc_fic = new CharSwr( fic_automate );
        int var_a_la_con=0;
        while ( var_a_la_con != -1 ) {
            base_1 = desc_fic.nextNum();
            base_2 = desc_fic.nextNum();
            base_3 = desc_fic.nextNum();
            base_4 = desc_fic.nextNum();

            if ( base_1 != -1 && base_2 != -1 && base_3 != -1 && base_4 != -1 ) {

                state = new Etat(base_1, base_2, base_3, base_4);
                states.add(state);
            }
            else
            {
                var_a_la_con = -1;
            }
        } //fin while
    } // fin du bloc try
    catch (FileNotFoundException e) {
        System.err.println("Fichier inexistant :" + e);
        throw(e);
    }
    catch (IOException e) {
        System.err.println("IOError:" + e);
        throw(e);
    }
} //fin constructeur

/**
 * methode renvoie le nombre total d'etats
 */
public int nbEtats () {

    return states.size();

}

/*
 * renvoie un etat en particulier
 * @param position les Etats sont contenus dans un vecteur
 */
public Etat getEtatAt( int position ) {
    return (Etat) states.elementAt(position);
}

/* Affiche le contenu de l'automate.
 * Methode de test intermediaire
 */
public void printAutomate() {
    for (int i=0; i < states.size() ; i++){
        state = (Etat) states.elementAt(i);
        state.printEtat() ;
    }

}

}

```

```
//classe Etat
public class Etat {
    static int i=0;
    private int G, C, T, A ;
    private int no_eta;

    //constructeur
    public Etat (int a, int b, int c, int d ){
        G=a-48;
        C=b-48; // pour avoir la val num
        T=c-48; // ascii de 0 est 48, 1 ~ 49 etc
        A=d-48; // bref
        no_eta = i;
        i++;
    }

    //methode
    public int getPosEtat () {
        return no_eta;
    }

    public int nextEtat ( int nucleotide ){
        int suivant= -1 ;
        switch (nucleotide) {
            case 103 : case 71 : //ascii de G et g
                suivant = G;
                break;
            case 67 : case 99 : //ascii de C et c
                suivant = C;
                break;
            case 84 : case 116 : //ascii de T et t
                suivant = T;
                break;
            case 65 : case 97 : //ascii de A et a
                suivant = A;
                break;
        }
        return suivant;
    }

    public void printEtat() {
        System.out.println("          GCTA");
        System.out.println ( "etat no " + no_eta + " matrix : " + (char) G + (char) C + (char) T + (char) A );
    }

}
```

```

////////////////////////////////////
//// CLASSE CharSwalower //
////////////////////////////////////
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.lang.*;
import java.io.*;

public class CharSwr {

private int position=0; // position de lecture
    private FileReader fic;
    ///////////////////////////////////
    /// CONSTRUCTEURS //
    ///////////////////////////////////

public CharSwr (String le_fic) throws FileNotFoundException {
    try {
        fic= new FileReader(le_fic);
    }
    catch (FileNotFoundException e) {
        System.err.println("fichier inexistant: " + le_fic.toString() );
        throw(e);
    }
}

    ///////////////////////////////////
    // Methodes //
    ///////////////////////////////////
public int nextNum() throws FileNotFoundException, IOException {

    int ascii;
    try {
        ascii = fic.read();
        position++;
    } //fin bloc try
    catch (FileNotFoundException e) {
        System.err.println("Fichier inexistant : " + e);
        throw(e);
    }
    catch (IOException e) {
        System.err.println("IOError:" + e);
        throw(e);
    }
    //System.out.println( "lu : " + (char) ascii + " >> a la position : " + position );
    return ascii;
}

public int nextChar() throws FileNotFoundException, IOException {
    int ascii;
    int valeur=0;
    try {
        ascii = fic.read();
        position++;
        if (ascii == 65 | ascii == 84 | ascii == 71 | ascii == 67 | ascii == 97 | ascii == 116 |
ascii == 103 | ascii == 99 ) {
            valeur = ascii;
        } else if (ascii == -1) {
            System.out.println("fin fichier");
            fic.close();
        } else {
            // System.err.println("fichier " + fic.toString() + " contient des
chars inatendus" );
        }
    }
    catch (FileNotFoundException e){
        System.err.println("Fichier inexistant : " + e);
        throw(e);
    }
    catch (IOException e) {
        System.err.println("IOError:" + e);
        throw(e);
    }
    return valeur;
}

public int getPosition() {
    return position;
}

public boolean isOpen() throws IOException {
    try {
        return fic.ready();
    } catch (IOException e) {
        System.err.println("IOError:" + e);
        throw(e);
    }
}
}

```

## GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats

include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

## O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of

Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may

distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.