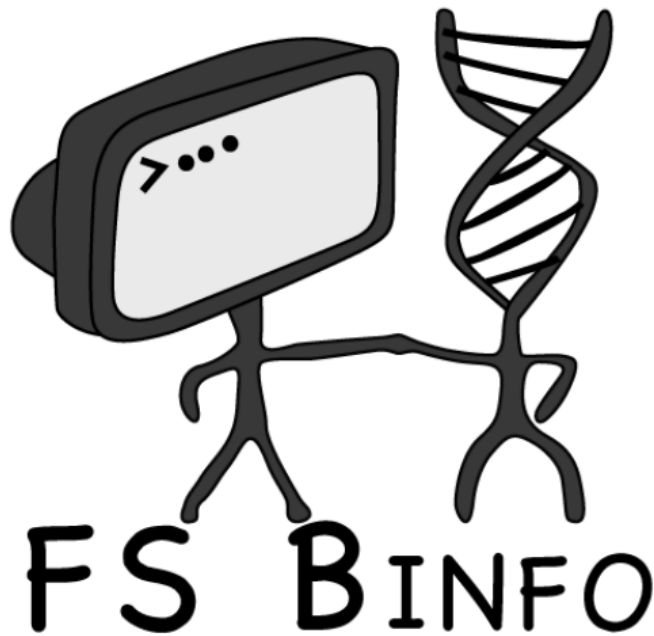


# Programmier-Aufgabe zur Wiederholung



Hans Hawe

Fachschaft Bioinformatik

21.12.2010

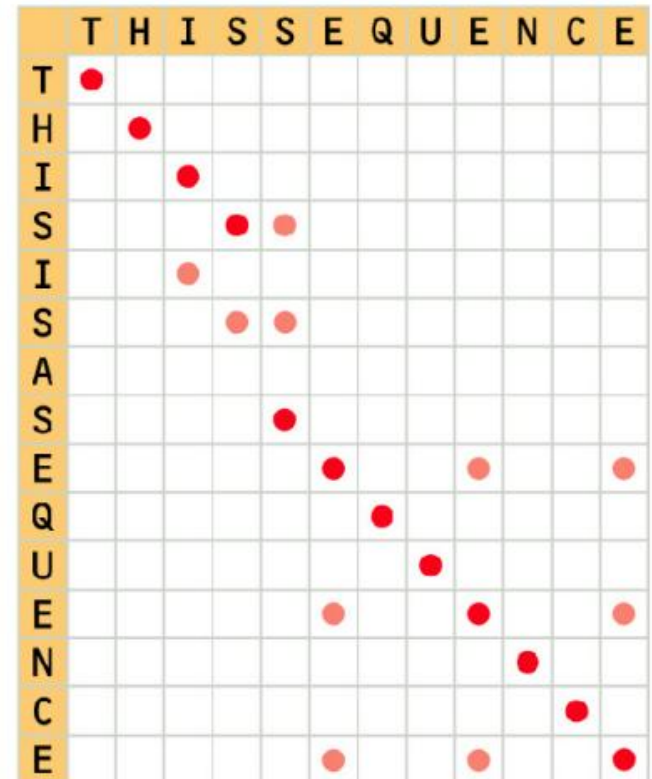
(Folien basierend auf Tutorium von  
Fabian Schmich, 15.12.2009)

# Wiederholung

- Variablen definieren
- Arrays
- Kontrollstrukturen (if/else, for/while)
- Objektorientierte Programmierung
- Methoden
- File In-/Output

# Aufgabe: Dotplot

- Was ist ein Dotplot?
  - Einfacher Sequenzvergleich
- Sequenzen an X- und Y-Achse
- Punkt bei übereinstimmenden Positionen
  - Identifizierung bestimmter Sequenzmerkmale



(Zvelebil & Baum)

# Anforderung

- Verarbeiten von 2 Sequenzen als Input
  - Evtl. Auslesen aus Dateien
- Berechnen der Hit-Matrix
- Ausgabe des Plots auf Console
  - Evtl. Ausgabe auch in Datei

# 1. Aufgabe: Konzept

- Benötigte Klassenvariablen?  
...
- Benötigte Methoden?  
...
- Welche Klassen werden benötigt?  
...

# Klasse: Dotplot

- Variablen:
  - private char hitSymbol;
  - private String seq1;
  - private String seq2;
  - private boolean[][] hitMatrix;
- Methoden:
  - Konstruktor(en)
  - calcHitMatrix();
  - getPlotChar();
  - printPlot(); (printToFile());
  - ReadSequence();

## 2. Aufgabe: Umsetzung

- Implementieren aller Methoden
- Erstellen einer main-Methode
- Testen der Funktionalität
- Evtl. Erweiterung: `printToFile();`

# Klasse Dotplot

```
1 public class Dotplot {
2     private String seq1 = null;
3     private String seq2 = null;
4     private boolean[][] hitMatrix;
5     private static char hitSymbol = 'x';
6
7     public Dotplot(String seq1, String seq2) {
8         this.seq1 = seq1;
9         this.seq2 = seq2;
10        this.hitMatrix =
11            new boolean[seq1.length()][seq2.length()];
12    }
```

# Methode getChar()

```
1 private char getChar(int x, int y) {  
2     if (this.hitMatrix[x][y]) {  
3         return this.hitSymbol;  
4     } else {  
5         return ' ';  
6     }  
7 }
```

# Methode PrintPlot()

```
1 public void PrintPlot() {
2     for (int i = -1; i < this.seq1.length(); i++) {
3         StringBuffer sb = new StringBuffer();
4         for (int j = -1; j < this.seq2.length(); j++) {
5             if (i == -1) {
6                 if (j == -1) {
7                     sb.append(" ");
8                 } else {
9                     sb.append(" " + this.seq2.charAt(j));
10                }
11            } else {
12                if (j == -1) {
13                    sb.append(this.seq1.charAt(i));
14                } else {
15                    sb.append(" " + this.getChar(i, j));
16                }
17            }
18        }
19        System.out.println(sb.toString());
20    }
21 }
```

# Methode CalcPlot()

```
1 public void CalcPlot() {
2     for (int i = 0; i < this.seq1.length(); i++) {
3         for (int j = 0; j < this.seq2.length(); j++) {
4             // charAt gets the char of a String at position i/j
5             if (this.seq1.charAt(i) == this.seq2.charAt(j))
6                 {
7                     this.hitMatrix[i][j] = true;
8                 } else {
9                     this.hitMatrix[i][j] = false;
10                }
11        }
12    }
```

# Methode ReadSequence()

```
1 public static final String ReadSequence(File seq) {
2     // create stringbuffer to store sequence
3     StringBuffer sequence = new StringBuffer();
4     try {
5         // default way to read files
6         FileReader fr = new FileReader(seq);
7         BufferedReader reader = new BufferedReader(fr);
8         String line = null;
9         // read until there is no more line
10        while ((line = reader.readLine()) != null) {
11            // append to current seq
12            sequence.append(line);
13        }
14        // !!
15        fr.close();
16    } catch (FileNotFoundException e) // handle file not found exception
17    {
18        System.err.println("Cannot find file: " + seq.getPath());
19    } catch (IOException ex) // handle default ioexception
20    {
21        System.err.println("Cannot read from file: " + seq.getPath());
22    }
23    return sequence.toString();
24 }
```

# Neuer Konstruktor

```
1 public Dotplot(File f1, File f2) {  
2  
3     this(Dotplot.ReadSequence(f1),  
          Dotplot.ReadSequence(f2));  
4 }
```

# Methode Main()

```
1 public static void main(String[] args) {
2     File f1 = null;
3     File f2 = null;
4     Dotplot d = null;

    // check arguments
5     if (args.length == 2) {
6         f1 = new File(args[0]);
7         f2 = new File(args[1]);
8         d = new Dotplot(f1, f2);
9     } else {
10        // create dotplot with random strings
11        d = new Dotplot("ABCDEFGHIBCEFGHI",
    "ABCDEFGHIBCEFGHI");
12    }
13    d.CalcPlot();
14    d.PrintPlot();
15 }
```

# Frohe Weihnachten!

