# MstatX User Manual

Guillaume Collet

May 13, 2011

# Contents

# 1 Introduction

MstatX provides a simple and easy-to-extend solution for multiple alignment scores calculation.

A multiple alignment can be produced in many ways, including the use of an application like CLUSTALW [5] or MUSCLE [1], but also manually. Once obtained, automatic methods gives a global score for the alignment. But we often need to calculate the conservation on one or many columns. Or we want to calculate a global score with another method or statistic.

When we searched the web for such a tool, we found nothing. Many scores exist to evaluate a multiple alignment, but no simple command-line tool can do it easily. That is why we created MstatX.

In MstatX, things are simple. A multiple alignment is simply defined by an alphabet of symbols and a list of words made with this alphabet. The calculation of a given score is completely independent from the other scores. The output is a simple text file with the score of each column of the multiple alignment (or the global score only if asked).

The default usage of MstatX is for multiple alignment of protein sequences, so, we implemented an "amino acids substitution matrix" reader. In fact, it can read any substitution matrix of any size, so it can be used for DNA scoring.

MstatX has two side: the end-user side and the programmer side. If you wants to calculate statistics already available in MstatX, then read the section 3. If you want to add a new statistics module in MstatX, then read the section 4 to have some examples of already coded statistics modules.

# 2 Installation

If you read this, then you have downloaded MstatX from my github repository. When you are at the root directory of MstatX, just open a terminal and:

```
make
```

This will compile MstatX with `g++`. So you need it in order to compile.

MstatX uses only one environment variable: `SCORE_MAT_PATH`. This variable gives the path to the substitution matrices directory. You can set this variable in your `.bashrc` file or you can also use the `-sp` option to specify the substitution matrices directory to MstatX (default substitution matrices are in `data` repertory).

# 3 User side

This section is dedicated to the usage of MstatX. If you want to add a statistics module in MstatX, read section 4.

## 3.1 Getting started

The basic use of MstatX is like the following:

```
mstatx -ma example.mali
```

This command will calculate the trident statistic of each columns in the multiple sequences alignment given in file `example.mali`. The scores are written in file `example.stat`. The score on line 1 of this file is the score of column 1, line 2 is the score of column 2, etc.

The trident statistic is defined in section 3.3.

## 3.2 Options

## 3.3 Available statistics

Statistics proposed in MstatX comes from many articles. For a review of these statistics, you can refer to [6] and [3]. Some of the statistics use scoring matrices. We choose to use matrices in the AAindex format [4] from the AAindex website[1]. Although it has not been updated since 2008, this website provide a useful list of amino acids scoring matrices in a simple format. The user can use his own scoring matrix by the flag -sp and -sc.

To illustrate the results of MstatX, we will use the following multiple alignment example:

### Gap

The Gap statistic is simply the proportion of gaps in columns.

[1] `http://www.genome.jp/aaindex/`

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| a | D | D | D | D | D | D | I | P | D | L |
| b | D | D | D | D | D | D | I | P | V | L |
| c | D | D | D | D | D | D | I | P | Y | L |
| d | D | D | D | D | D | D | I | P | A | - |
| e | D | D | D | D | D | D | L | W | T | - |
| f | D | D | E | D | E | E | L | W | K | - |
| g | D | D | E | D | E | E | L | W | P | - |
| h | D | D | E | D | E | F | V | S | R | - |
| i | D | E | E | F | F | F | V | S | H | - |

Table 1: Multiple alignment example from Valdar [6]. This multiple alignment is used to illustrates the available statistics in MstatX.

**Trident**

The trident statistic module is based on the work of William S.J. Valdar [6]. It is composed of three parts, each measures a different aspect of column conservation.

The first part, noted $t(x)$, measures the entropy of a column $x$ by the Shannon formula:

$$t(x) = \frac{1}{log(min(N, K))} \sum_{a=1}^{K} p_a log(p_a) \tag{1}$$

In this formule, $N$ is the number of sequences in the multiple alignment, $K$ is the size of the alphabet, and $p_a$ is the probability of symbol $a$ in the column $x$. The redondancy between the sequences in the multiple alignement is measured in $p_a$ by this formula:

$$p_a = \sum_{i \in \{i|s(i)=a\}} w_i \tag{2}$$

In this formula, $w_i$ is the weight of sequence $i$ and it is added to the probability $p_a$ only if the symbol of sequence $i$ and column $x$ is $a$. The weight is calculated by the formula from Henikoff & Henikoff [2]:

$$w_i = \frac{1}{L} \sum_{x=1}^{L} \frac{1}{K_x n_{x_i}} \tag{3}$$

In this formula, $L$ is the length of the alignment,

# 4 Programmer side

MstatX tries to provide an easy-to-extend application for scoring a multiple alignment. So MstatX keeps its simplicity even in the code.

The main.cpp file contains the main function. This function simply:

- reads the command line;
- initialize the statistic factory;
- reads the multiple alignment file;
- calculate the statistic;
- print the results in output;
- print some informations on standard output.

This simplicity is provided by the virtual class Statistic which allows the use of a factory design pattern to create the right sub-class. The virtual class Statistic is described in the next section.

## 4.1 The Statistics virtual class

If you want to add a statistic calculation, you need to write a class inherited from the Statistic class.

```
class Statistic {
  public:
    Statistic(){};
    virtual ~Statistic(){};
    virtual void calculateStatistic(Msa & msa){};
    virtual void printStatistic(Msa & msa){};
};
```

In the Statistic class, only two methods are virtual: calculateStatistic(Msa & msa) and printStatistic(Msa & msa). So you have to rewrite only two methods to create a new statistic.

# Bibliography

[1] Robert C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[2] Steven Henikoff and Jorja G. Henikoff. Position-based sequence weights. *Journal of Molecular Biology*, 243(4):574 – 578, 1994.

[3] Fredrik Johansson and Hiroyuki Toh. A comparative study of conservation and variation scores. *BMC bioinformatics*, 11:388, 2010.

[4] Shuichi Kawashima, Piotr Pokarowski, Maria Pokarowska, Andrzej Kolinski, Toshiaki Katayama, and Minoru Kanehisa. Aaindex: amino acid index database, progress report 2008. *Nucleic Acids Research*, 36(suppl 1):D202–D205, 2008.

[5] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins. Clustal w and clustal x version 2.0. *Bioinformatics*, 23(21):2947–2948, 2007.

[6] W.S.J Valdar. Scoring residue conservation. *Proteins: Structure, Function, and Bioinformatics*, 48(2):227–241, 2002.

# Licence