XXXVI Ibero-Latin American Congress on Computational Methods in Engineering
CILAMCE
Rio de Janeiro, 22-25 Nov 2015

# CLASSIFICAÇÃO DE FITOPLÂNCTONS EM AMBIENTES AQUÁTICOS DA COSTA DO ESTADO DO RIO DE JANEIRO

**Jonathan R. Porto, Matheus H. K. Galvez**

**Daniele B. Peçanha, Matheus C. Ambrozio**

**Alexandre S. P. Miloski, Valeria M. Bastos**

{jrp2308, mhkgalvez, danielepecanha, mambrozio, valeriab}@dcc.ufrj.br

alexandre.miloski.ufrj@hotmail.com

DCC/IM/UFRJ – Departamento de Ciência da Computação, Universidade Federal do Rio de Janeiro

Avenida Athos da Silveira Ramos, 274, Cidade Universitária, Rio de Janeiro, Brasil

**Myrian C. A. Costa, Nelson F. F. Ebecken**

{nelson, myrian}@ntt.ufrj.br

NTT/COPPE/UFRJ – Núcleo de Transferência de Tecnologia, Universidade Federal do Rio de Janeiro

Centro de Tecnologia, Bloco B, Cidade Universitária, Rio de Janeiro, Brasil

***Resumo.*** *Processos de identificação de padrões têm se deparado cada vez mais com o problema de análise de grandes volumes de dados, coletados através de sensores ou procedimentos de laboratório. Este trabalho propõe o desenvolvimento de uma metodologia de classificação de espécimes de fitoplânctons utilizando espécies previamente classificadas, coletadas através de leituras óticas realizadas por um citômetro de fluxo localizado na costa do estado do Rio de Janeiro. A classificação foi realizada com o uso do algoritmo k-NN que utilizou como conjunto de treinamento medidas de dissimilaridade retornadas pelo algoritmo Dynamic Time Warping (DTW). A técnica de restrição por janelas proposta por Sakoe e Chiba [1] foi incorporada ao DTW com o objetivo de diminuir seu tempo de execução. Os resultados da classificação em ambos os casos foram semelhantes, porém o tempo de execução do DTW foi menor quando aplicada a janela.*

***Palavras-chave:*** *Classificação, Aprendizado de Máquina, Dissimilaridade, Ecossistemas Aquáticos, k-NN*

# 1 INTRODUÇÃO

Processos de identificação de padrões têm se deparado cada vez mais com o problema de análise de grandes volumes de dados devido ao aumento da capacidade e quantidade de sensores que coletam informações relevantes. Nem sempre é trivial extrair informações desses volumes de dados e lidar com um alto custo computacional.

É possível realizar o monitoramento de sistemas aquáticos a partir de dados coletados por citômetros de fluxo [2] [3]. O processamento desses dados pode auxiliar na definição do estado trófico e nível de qualidade de ecossistemas aquáticos. Além disso, o gerenciamento de zonas costeiras requer que a análise desses dados seja realizada em tempo hábil para a tomada de decisões, o que torna esta área um campo fértil para aplicação de técnicas de descoberta de conhecimento.

O uso de algoritmos de mineração de dados permite explorar dados e evidenciar padrões normalmente não evidentes e, dessa forma, auxiliar na descoberta de conhecimento (padrões no comportamento ou na variabilidade ambiental) [4].

Este trabalho propõe o desenvolvimento de metodologias computacionais para análise de dados heterogêneos, compostas de assinaturas óticas de espécimes de fitoplânctons coletadas através de um citômetro de fluxo, localizado na costa do estado do Rio de Janeiro, com o objetivo de classificar esses espécimes, auxiliando na avaliação da qualidade dos ecossistemas aquáticos. Para realizar esta avaliação, foi desenvolvido um classificador de espécimes de fitoplânctons a partir de dados manualmente classificados.

Na seção 2, será descrita a técnica de citometria de fluxo e como ela foi utilizada para o trabalho proposto. A seção 3 descreve o algoritmo Dynamic Time Warping, DTW, conforme proposto por Sakoe e Chiba [1], utilizado para cálculos de dissimilaridade, além de uma versão modificada que permite sua aplicação aos dados do citômetro. Na seção 4 será apresentado o método de classificação em si, realizado pelo algoritmo k-NN. Na seção 5 serão apresentados diversos estudos de caso e o ambiente computacional utilizado para tentar minimizar o elevado tempo de execução gerado devido ao expressivo número de espécimes. Na seção de conclusão, será apresentada a avaliação dos resultados obtidos.

# 2 METODOLOGIA

Citômetros de fluxo medem o espalhamento luminoso e a fluorescência das partículas que passam por uma zona luminosa (laser) muito intensa, transportadas por um jato de água com alta velocidade (2 m/s), em uma cubeta de quartzo para o fluxo. As partículas em suspensão (células) são bombeadas em fila única através do ponto de análise, tipicamente 1000 células por segundo, com uma velocidade analítica de 3 a 5 minutos por amostra a cada 10 minutos. Os sucessivos sinais de feixes luminosos espalhados e fluorescências geradas por cada partícula são detectados por tubos fotomultiplicadores ou fotodiodos. A sensibilidade de detecção é suficiente para analisar partículas micrométricas. A interface eletrônica converte estes sinais primários em dados digitais que são armazenados em um disco rígido para análise, realizando contagens citométricas precisas e confiáveis.

O citômetro de fluxo utilizado [3] realiza as leituras dos cinco descritores das células que passam em seu interior por meio de cinco sensores óticos. Esses descritores correspondem às fluorescências vermelha (FLR), que caracteriza pigmentos de clorofila, amarela (FLY) e laranja (FLO), e ao tamanho e estrutura interna da célula, respectivamente FWS (Forward

Scatter) e SWS (Sideward Scatter). Os sinais captados em cada uma das leituras são, então, armazenados em um formato de arquivo proprietário, podendo ser lido apenas pelo software *Cytoclus* [6]. Este software foi desenvolvido pelo fabricante do citômetro de fluxo *Citobuoy* [6] e permite a conversão dos dados para formatos abertos e portáveis. Todo este processo é conhecido como citometria de fluxo [2] [5], e o funcionamento interno do citômetro é ilustrado na Fig. 1.

A metodologia consiste na construção de um classificador de células de fitoplânctons a partir de dados de vinte e duas espécies previamente classificadas por biólogos em laboratório. Para automatizar esta tarefa, a comparação visual foi traduzida em medidas de dissimilaridade, tornando possível sua execução através de métodos computacionais. O algoritmo utilizado para gerar essas medidas foi o DTW, conforme proposto em [1]. Outro método, este proposto em [7], tem como objetivo classificar células desconhecidas enquanto que a metodologia aqui proposta visa, como prova de conceito, confrontar dados classificados com dados já previamente identificados.

O algoritmo de classificação escolhido foi o k-NN, que recebeu como entrada as medidas de dissimilaridade geradas pelo DTW. Os resultados de classificação são discutidos na seção 5.

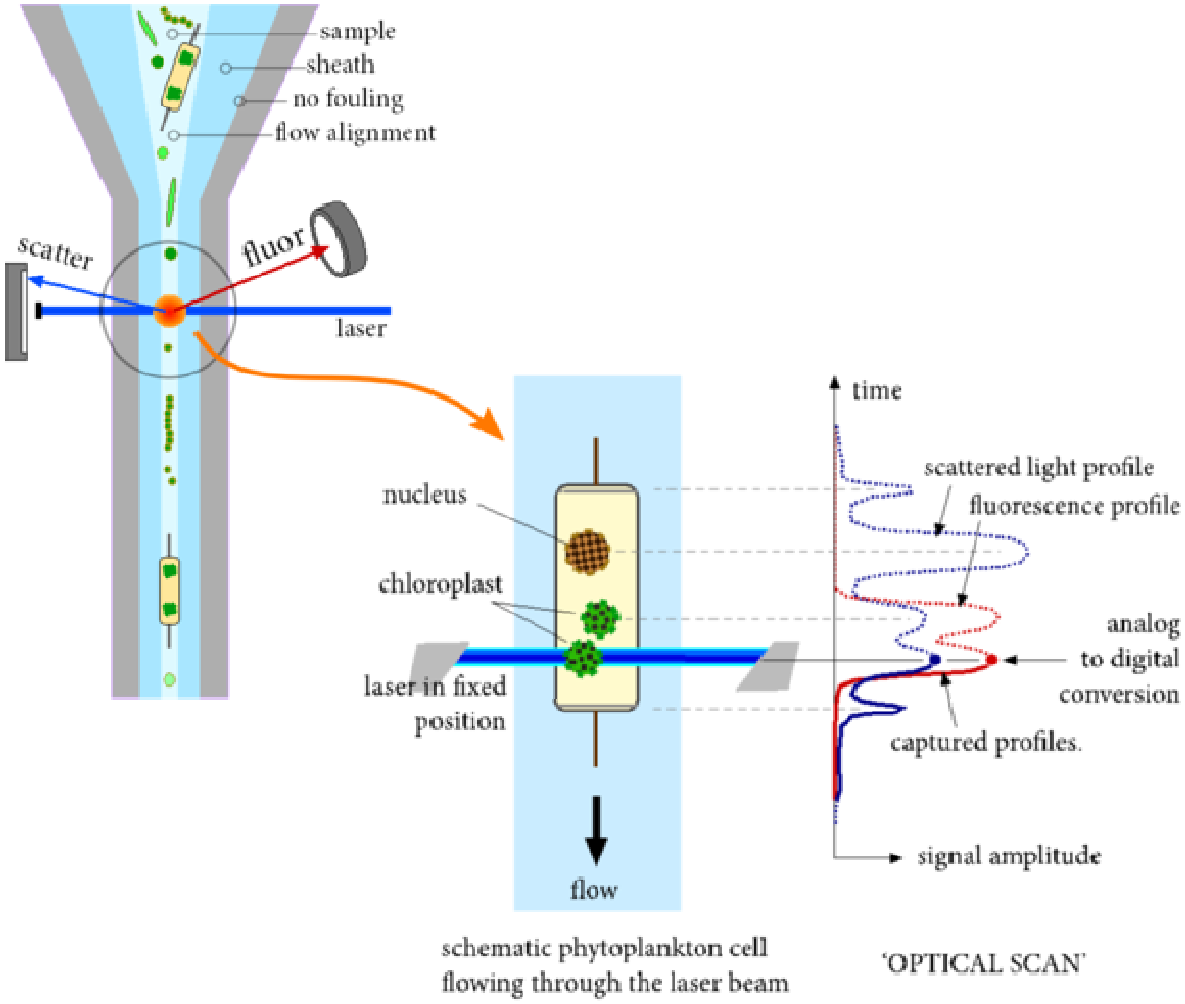


**Figura 1. Citometria de fluxo. Imagem referente ao citômetro *Cytobuoy* [6].**

# 3 CÁLCULO DE DISSIMILARIDADE

## 3.1 Dynamic Time Warping

O algoritmo DTW é um método de programação dinâmica, proposto por Sakoe e Chiba [1], para aplicação em reconhecimento de padrões de fala através da correspondência entre duas sequências temporais e do cálculo da distância entre elas; esta distância representa o custo da correspondência entre as sequências.

Como mostrado na seção 4, o processo de classificação adotado baseia-se nas diferenças entre as sequências que representam as células de fitoplânctons. O custo da correspondência entre duas sequências, fornecida pelo DTW, é interpretada neste trabalho como a dissimilaridade entre as células, justificando, assim, seu uso para a geração dos dados de entrada do classificador k-NN. O DTW é descrito no Algoritmo 1 a seguir.

**Algoritmo 1 -** Algoritmo DTW para computar distância acumulada.

```
DistânciaDTW (s: vetor(1..n), t: vetor(1..m))
Inicio
    DTW <- matriz(0..n, 0..m)

    De i <- 1 Até n Faça
        DTW(i, 0) <- Infinito
    Fim

    De i <- 1 Até m Faça
        DTW(0, i) <- Infinito
    Fim

    DTW(0, 0) <- 0

    De i <- 1 Até n Faça
        De j <- 1 Até m Faça
            custo <- Distância(s(i), t(j))
            DTW(i, j) <- custo + Mínimo(DTW(i-1, j), DTW(i, j-1),  DTW(i-1,
j-1))
        Fim
    Fim

    Retorna elemento de índice (n, m) da matriz DTW
Fim
```

O DTW calcula a distância entre os pontos das sequências pela fórmula euclidiana, representada, no Algoritmo 1, pela função *Distância(i, j)*. A expressão dessa fórmula é dada na equação (1).

$$Distância(x, y) = |x - y| \,. \tag{1}$$

Como é possível notar no algoritmo, sua complexidade é $O(n*m)$, se $n$ e $m$ representam os comprimentos das sequências, e, caso os comprimentos sejam idênticos, o algoritmo passa a ter complexidade $O(n^2)$. No estudo de caso apresentado na seção 5, centenas de milhares de espécimes são comparadas, cada uma com todas as demais, a fim de gerar um modelo para o algoritmo de classificação. Visto que cada comparação é realizada com uma chamada do algoritmo DTW, de complexidade quadrática, o tempo total de execução de todas as

comparações pode ser grande ao ponto de tornar impraticável sua execução num computador comum. Para minimizar o tempo de execução, este trabalho utiliza-se de um ambiente de computação distribuído onde as comparações entre os espécimes (execuções do DTW) tem sua execução acelerada pela maior capacidade de processamento do sistema. Este ambiente também é descrito na seção 5.

Cada espécime de fitoplâncton é descrito por uma sequência de tuplas de cinco valores, que representam seus cinco descritores, o que faz com que cada um deles tenha dimensão cinco. Para se obter a dissimilaridade entre sequências deste tipo (multidimensionais) foi utilizada a versão multidimensional do DTW.

## 3.2 DTW Multidimensional

A diferença fundamental entre a versão anterior e a versão multidimensional reside no fato de que, ao invés do custo calculado ser um valor escalar, ele é, agora, representado por uma matriz. Como entrada, o DTW multidimensional recebe duas séries de dimensão C, com *m* e *n* linhas, respectivamente. A matriz de distâncias, de dimensão *n* X m é calculada pela equação (2). O custo de cada *loop* no Algoritmo 1 foi calculado pelo acesso à matriz de distâncias, somente nas posições *i, j*. É importante notar, ainda, que a versão multidimensional do algoritmo adiciona uma carga de processamento, representada pelo o cálculo da matriz de distâncias euclidianas.

$$Distância(i,j) = \sum_{k=1}^{C} |s(i,k) - t(j,k)| \tag{2}$$

## 3.3 Restrição de Janela no Cálculo da Distância DTW

Conforme é possível observar, o DTW é um algoritmo de programação dinâmica de complexidade $O(n^2)$. Estudos adicionais, como o método Sakoe e Chiba Band [1], apresentam alterações que possibilitam a diminuição do tempo de execução do DTW. O método Sakoe e Chiba Band prevê a existência de uma janela de ajuste (*Adjustment Window*) durante a execução do algoritmo DTW, conforme ilustrado na Fig. 2.

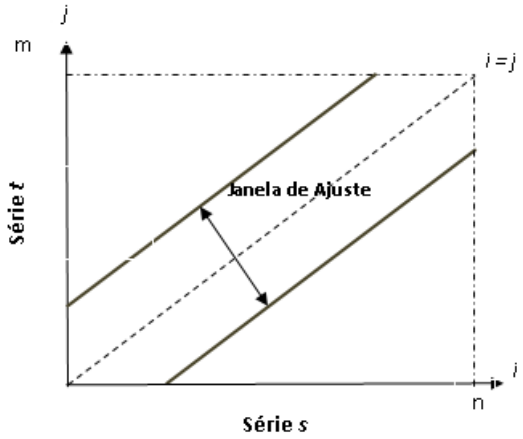


**Figura 2. Janela de ajuste, conforme proposto em [1]**

A matriz DTW é preenchida enquanto o algoritmo a percorre, desde a posição (1, 1) até a posição (*n*, *m*). Neste percurso, cada posição (*i*, *j*) depende do valor mínimo entre as três posições adjacentes (*i-1, j-1*), (*i*, *j-1*) e (*i-1*, *j*).

A proposta em [1] consiste em aplicar uma restrição à execução do DTW de modo a não permitir que o algoritmo ultrapasse a janela de ajuste quando preenche qualquer posição da matriz. Isto é feito adicionando-se uma condição ao preenchimento da matriz DTW no Algoritmo 1, de modo a verificar se a posição atual da iteração está dentro ou fora da janela. Caso a posição esteja fora da janela, ela é ignorada e a iteração segue adiante. Isto evita que os cálculos de distância e de verificação de valor mínimo sejam feitos, acelerando a execução do algoritmo. O Algoritmo 2, abaixo, apresenta a versão modificada do DTW com a implementação deste método.

**Algoritmo 2 -** Algoritmo DTW modificado (*Sakoe e Chiba Band*).

```
DistânciaDTW (s: vetor(1..n), t: vetor(1..m))
Inicio
    DTW <- matriz(0..n, 0..m)

    De i <- 1 Até n Faça
        DTW(i, 0) <- Infinito
    Fim

    De i <- 1 Até m Faça
        DTW(0, i) <- Infinito
    Fim

    DTW(0, 0) <- 0

    De i <- 1 Até n Faça
        De j <- 1 Até m Faça
            Se (i, j) está dentro da janela de ajuste Então
                custo <- Distância(s(i), t(j))
                DTW(i, j) <- custo + MinimoModificado(DTW(i-1, j), DTW(i,
j-1),  DTW(i-1, j-1))
            Fim
        Fim
    Fim

    Retorna elemento de índice (n, m) da matriz DTW
Fim
```

É importante observar, no Algoritmo 2, que algumas posições ficam de fora do cálculo de custo, caso a condição do teste (em negrito no algoritmo) seja falsa. Assim, a janela deve ter um tamanho grande o suficiente para que a última posição do vetor não seja ignorada, uma vez que esta será o custo retornado pelo algoritmo.

Outra mudança importante é a função *MinimoModificado()*. Esta modificação no cálculo do valor mínimo é necessária para que apenas posições dentro da janela de ajuste sejam computadas. Assim, esta função irá desconsiderar de seu cálculo qualquer um dos três valores passados como parâmetro cujo índice esteja fora da janela de ajuste.

A técnica Sakoe e Chiba Band pode introduzir distorções no cálculo da distância DTW quando o algoritmo calcula o menor valor dentre três posições adjacentes a uma determinada célula (*i*, *j*) da matriz. Isto ocorre porque, eventualmente, a posição que contém o valor mínimo, pode ser desconsiderada por estar fora da janela de ajuste e, sendo assim, outro valor será considerado como mínimo e utilizado no cálculo, de modo que o valor final da dissimilaridade entre duas séries sofrerá alterações. Na Fig. 3, esta distorção é bem ilustrada.

A célula destacada de cinza, de índice (*i-1*, *j-1*), se encontra fora da janela de ajuste. Sendo assim, o algoritmo não irá considerá-la no cálculo da distância mínima.

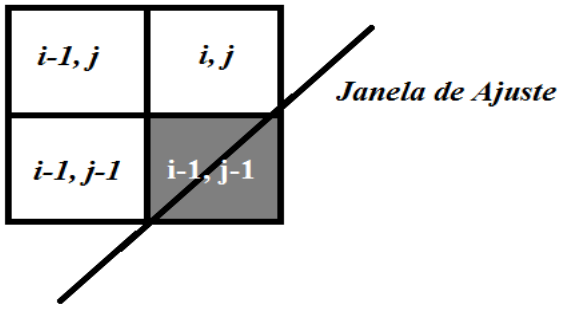


**Figura 3. Eliminação da célula (i-1, j-1) pela técnica da janela de ajuste.**

Além de evitar que mais processamento seja gasto com cálculos desnecessários, esta técnica pode, ainda, atingir um custo amortizado *O(n),* dependendo do tamanho escolhido para a janela de ajuste, conforme indicado em [8]. O trabalho em [8] sugere que, além de aumentar o desempenho, essa técnica pode influenciar, também, positiva ou negativamente no resultado da classificação. O grau de influência sofrido será determinado pelo tamanho de janela aplicado, uma vez que o resultado da classificação depende dos valores obtidos pelo algoritmo DTW. Assim, de forma empírica, a classificação pode ser melhorada se um tamanho apropriado de janela for escolhido. O tamanho ótimo da janela deve ser determinado empiricamente para cada aplicação, e diferentes bases de dados podem apresentar tamanhos ótimos de janela diferentes.

# 4 CLASSIFICAÇÃO

O classificador foi construído com o algoritmo k-NN, utilizando o DTW para cálculo do custo de correspondência, ou distância, entre as sequências. O k-NN classifica cada célula comparando-a com seus *k* vizinhos mais próximos [4].

Como o k-NN é um algoritmo de classificação cujo aprendizado é baseado em instâncias, as características dessas instâncias são, portanto, determinantes no resultado final do algoritmo. A dissimilaridade entre os espécimes foi a característica escolhida para orientar a execução do algoritmo. Além disso, conforme indicam [9] e [10], o uso do DTW como função de custo para o k-NN, em geral, produz resultados mais acurados.

Neste trabalho, o processo de classificação das células foi realizado utilizando-se o k-NN com K=1, 10, 50, 100 e 200. O classificador foi treinado com um modelo contendo as distâncias entre todas as células do conjunto de treinamento, calculadas pelo DTW. Além disso, foram gerados dois modelos, um sem a utilização da técnica de Sakoe e Chiba e o outro utilizando a menor janela possível, cujo tamanho é $|m - n|$, sendo $m$ e $n$ os tamanhos das séries $t$ e $s$ respectivamente. Qualquer janela de tamanho menor que este, faria com que a célula final da matriz DTW, de índice $i = n$ e $j = m$, que representa o resultado do DTW, ficasse fora da janela de ajuste, conforme indicado na seção 3.3.

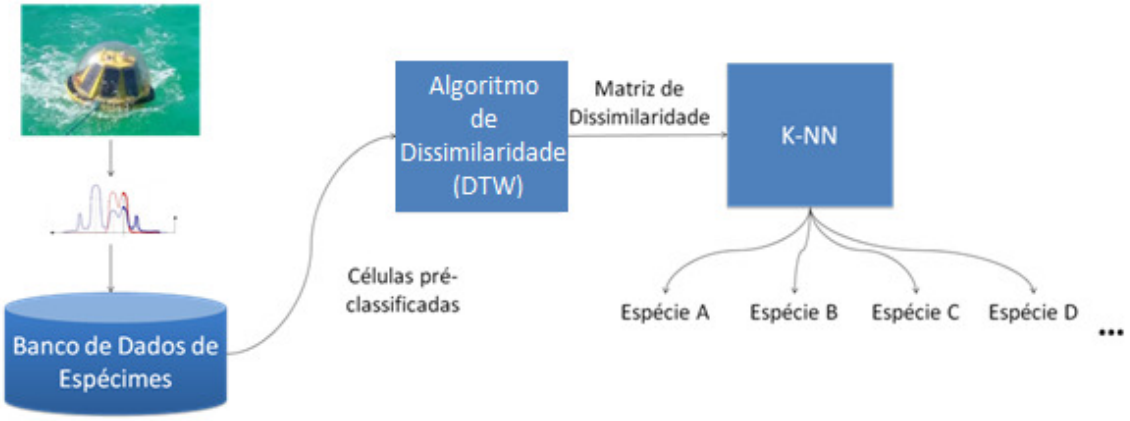


**Figura 4. Processo de construção do classificador.**

# 5 ESTUDO DE CASO E RESULTADOS

A base de dados utilizada nos experimentos é formada por informações de células de 22 espécies de fitoplânctons, totalizando 40.426 espécimes identificados. Parte desses dados foi utilizada como conjunto de treinamento para o classificador, a saber, 34.400 espécimes; desses dados foram selecionadas as 5 espécies mais representativas no ecossistema analisado, totalizando 1.536 células para testes.

Foram realizados dois tipos de teste: um sem a aplicação da restrição de Sakoe e Chiba e outro aplicando essa restrição com o menor valor possível para o tamanho de janela.

## 5.1 Ambiente Computacional

O cálculo de dissimilaridade e a classificação das células do conjunto de testes foram realizadas num ambiente formado por dois *clusters.* Um deles foi utilizado para implementação e teste dos algoritmos DTW e k-NN e o outro foi utilizado para produção, isto é, para o treinamento do classificador, utilizando-se os dados de 5 espécies.

O *cluster* de desenvolvimento é composto por 20 núcleos, 80GB de memória RAM agregada e armazenamento útil de 2.5TB, distribuídos em 5 máquinas, cada uma com um processador Intel i5-4460 (4 núcleos), 16GB de memória RAM e 1 disco rígido de 500GB. Já o *cluster* de produção é composto de 80 núcleos, 640GB de memória RAM agregada e armazenamento útil de 1.8TB para usuários e 7.2TB para o HDFS, distribuídos em 5 máquinas, cada uma com dois processadores Intel Xeon E5-2650 (8 núcleos), 128GB de memória RAM e 3 discos rígidos de 1TB. Ambos os clusters possuem instalado o sistema operacional Linux CentOS 6.5. O esquema dos *clusters* de desenvolvimento e produção são ilustrados nas Fig. 5a e 5b.

Ambos os *clusters* possuem o pacote R [11] instalado, utilizado para geração do modelo do DTW. O R é um pacote de computação numérica voltado para estatística, e possui diversas bibliotecas adicionais que podem ser acopladas a ele, proporcionando maior poder de processamento. Para que os diversos núcleos dos *clusters* pudessem ser melhor aproveitados, foi utilizada a biblioteca *snowfall* [12]. Atualmente, a *snowfall* é utilizada para dividir o processamento das matrizes de dissimilaridade, delegando-se um núcleo para o

processamento de cada espécie.

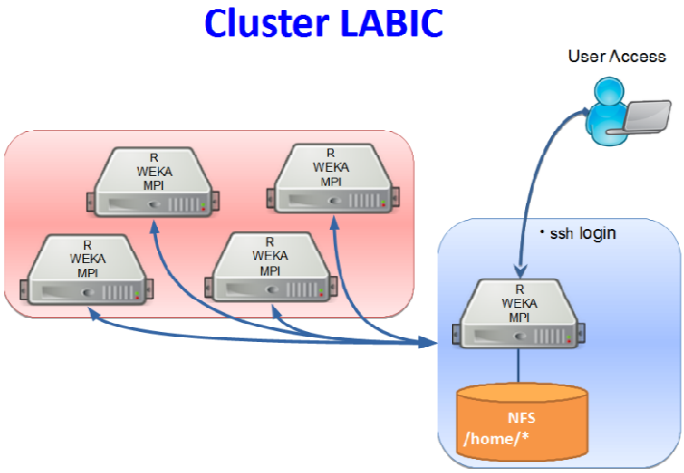


**Figura 5a. *Cluster* de desenvolvimento (LABIC).**

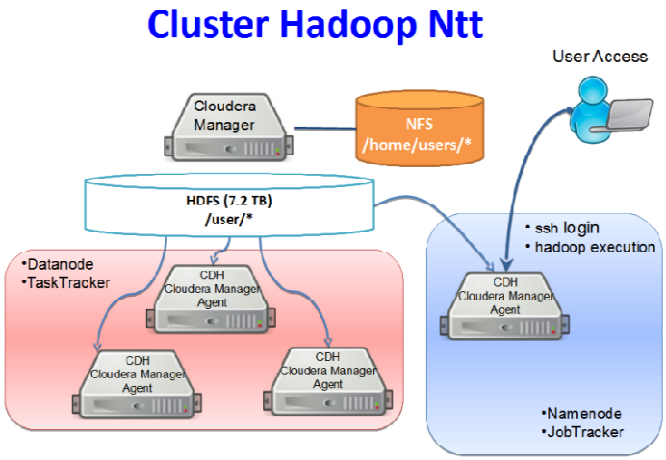


**Figura 5b. *Cluster* de produção (NTT/COPPE/UFRJ).**

## 5.2 Resultados

A geração dos modelos de treinamento do k-NN foi realizada no *cluster* de produção descrito na seção 5.1. As Tabelas 1 e 2 a seguir contêm os resultados da classificação para as espécies em ambos os casos, sem janela e com janela respectivamente. Para cada valor de k, o maior valor de classificação foi destacado. Deve-se destacar que a medida que os melhores valores para k, na maioria das espécies está entre 10 e 100. Uma pesquisa mais detalhada em relação a este parâmetro deve ainda ser explorada por trabalhos futuros.

**Tabela 1. Resultado da classificação do conjunto de teste (sem janela).**

| Espécie | K = 1 | K = 10 | K = 50 | K = 100 | K = 200 |
|---|---|---|---|---|---|
| BtIso | 32,45% | 46,92% | **54,82%** | **54,60%** | 51,97% |
| ChaetocerosGracilis | 23,88% | 27,53% | 22,26% | 23,48% | 24,29% |
| ChaetocerosTennueCB | 30,50% | 31,77% | 36,86% | 35,59% | 34,32% |
| Chlamidomonas | **41,05%** | 43,57% | 48,36% | 51,38% | 54,15% |
| Microcystis | 11,00% | **82,50%** | 23,50% | 36,50% | **56,50%** |

**Tabela 2. Resultado da classificação do conjunto de teste (com janela).**

| Espécie | K = 1 | K = 10 | K = 50 | K = 100 | K = 200 |
|---|---|---|---|---|---|
| BtIso | 30,26% | 43,42% | 49,56% | 51,31% | 46,49% |
| ChaetocerosGracilis | 22,26% | 27,93% | 24,69% | 23,07% | 20,24% |
| ChaetocerosTennueCB | 30,93% | 28,81% | 36,86% | 40,67% | 40,67% |
| Chlamidomonas | **44,08%** | 39,79% | **50,62%** | **52,89%** | 58,18% |
| Microcystis | 7,50% | **82,50%** | 17,50% | 30,50% | **60,00%** |

Os gráficos da Fig. 6 abaixo representam os percentuais de acerto da classificação das espécies para cada valor de K com e sem a utilização da técnica de restrição por janelas. Estes resultados mostram que o tamanho de janela adotado não causou impacto significativo na classificação para a base de dados utilizada.

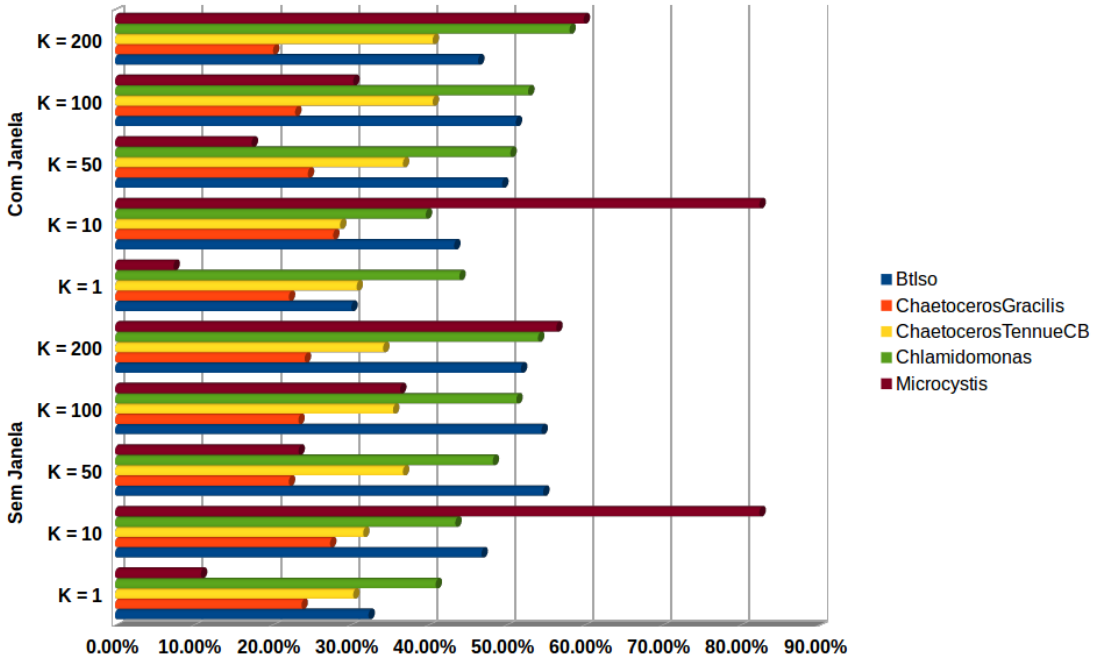


**Figura 6. Percentuais de acerto.**

# 6 CONCLUSÃO E TRABALHOS FUTUROS

O desenvolvimento de uma metodologia de classificação de partículas a partir de um conjunto reduzido de espécimes previamente identificadas foi apresentado. O algoritmo k-NN gerou um modelo de classificador que pode ser utilizado para a classificação das partículas de teste. Uma variação do algoritmo tradicional com a utilização da técnica de restrição de Sakoe

e Chiba para dois tamanhos de janela, o maior (sem restrição) e o menor, não causou impacto significativo na classificação, porém o tempo de execução do algoritmo diminuiu. Os tamanhos ideais para o valor de k ficaram entre 10 e 100.

Como trabalhos futuros, destacam-se a realização de testes para determinar um tamanho de janela que maximize a acurácia da classificação e o aprimoramento do classificador para reconhecer células não identificadas coletadas pelo citômetro em tempo hábil. A determinação de um valor de k também poderá ser melhor avaliada.

Além disso, é possível reduzir o tempo de geração do modelo ainda mais. Para isso, será necessário um ajuste na distribuição de carga de trabalho entre os nós do *cluster*, permitindo aos nós que, porventura, finalizem seu processamento possam realizar tarefas ainda pendentes. O desenvolvimento de um algoritmo com técnicas de processamento paralelo ou computação heterogênea já se encontra em avaliação para a melhora da performance geral do classificador.

## AGRADECIMENTOS

Os autores são gratos à FAPERJ pelo suporte financeiro a esta pesquisa.

## REFERÊNCIAS

*[1]* Sakoe, H., & Chiba, S., 1978. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, vol. 26, n. 1, pp. 43-49.

*[2]* Dubelaar G.B.J. & GERRITZEN P.L., 2000. CytoBuoy: A Step Forward Towards Using Flow Cytometry in Operational Oceanography, *Scientia Marina*, vol. 64, pp. 255-265.

*[3]* Dubellar, B. J., Greerders, P. J. F., 2004. Innovative Technologies to Monitor Plankton Dynamics. Scanning Flow Cytometry: A New Dimension in Real-Time, In-Situ Water Quality Monitoring. *Sea Technology*, pp. 15-21.

*[4]* Witten, I. H., Frank, E., Hall, M. A., 2011. Data Mining: Practical Machine Learning Tools and Techniques - 3rd. Ed., Morgan Kaufmann Series.

*[5]* Malkassian A., Nerini D., van Dijk M.A., Thyssen M., Mante C., & Gregori G., 2011. Functional Analysis And Classification Of Phytoplankton Based On Data From An Automated Flow Cytometer. *Cytometry Part A*, vol. 79A, pp. 263–275.

*[6]* CytoBuoy flow cytometry solutions. Disponível em http://www.cytobuoy.com/.

*[7]* Caillault, E., Hébert, P., & Wacquet, G., 1980. Dissimilarity-based Classification of Multidimensional Signals by Conjoint Elastic Matching: Application to Phytoplanktonic Species Recognition. *Engineering Applications of Neural Networks Communications in Computer and Information Science*, vol. 43, pp. 153-164.

*[8]* Ratanamahatana, C. A., & Keogh, E., 2004. Three Myths About Dynamic Time Warping in *Proc. 10th SDM.*

*[9]* Kurbalija, V., Radovanovi, M., Geler, Z., & Ivanović, M., 2014.The Influence of Global Constraints on Similarity Measures for Time-Series Databases. *Knowledge-Based Systems*, vol. 56, pp. 49-67.

*[10]* Bagnal, A., Lines, J., 2014. An experimental evaluation of nearest neighbour time series classification. Department of Computing Sciences, University of East Anglia, Norwich, UK, Tech. Rep. CMP-C14-01, 2014.

*[11]* The R Project for Statistical Computing. Disponível em https://www.r-project.org.

*[12] snowfall*: Easier cluster computing (based on snow). Disponível em https://cran.r-project.org/web/packages/snowfall/index.html.