

Organização Automática de Coleções de Partituras Musicais Utilizando Agrupamento Hierárquico

Lucas M. de Oliveira¹, André Y. Kashiwabara¹, Carlos N. Silla Jr.¹

¹Laboratório de Computação e Tecnologia Musical
Universidade Tecnológica Federal do Paraná
Av. Alberto Carazzai, 1.640 – 86300-000
Cornélio Procópio, PR, Brazil

Abstract. *This work proposes a system to automatically organize music sheets in MusicXML format using a music sequence comparison algorithm and the hierarchical clustering technique. Our experimental results show that this approach is often able to group together music scores from the same music genre.*

Resumo. *Neste trabalho foi proposto um sistema para organizar automaticamente partituras musicais em formato MusicXML utilizando um algoritmo de comparação de sequências musicais e a técnica de agrupamento hierárquico. Os resultados experimentais obtidos mostram que o sistema é capaz de agrupar partituras de um mesmo gênero musical.*

1. Introdução

De forma geral, a música pode ser entendida como a arte de organizar e combinar sons no espaço e no tempo [Priolli, 2007]. A cadência de acordes é um exemplo de organização espacial, enquanto que as notas de uma melodia configuram uma organização temporal [Miranda, 2002]. Uma vez que a música possui papéis importantes dentro dos contextos sociais e psicológicos, formas de representação eficazes devem permitir buscas por conteúdo que contemplem estes aspectos. Tais formas são tipicamente baseadas em quatro tipos de informações: gênero, emoção, estilo e similaridades [Huron, 2000].

Estas informações estabelecem as principais aplicações em pesquisas que envolvem recuperação da informação musical. Gêneros e estilos musicais são, portanto, importantes descritores, uma vez que sumarizam características (ou padrões) comuns entre peças. Estes descritores são usados há anos para organizar coleções e refletem interação entre culturas [Holt, 2007]. Apesar do amplo uso, gêneros musicais não formam um conceito claro e definido, e suas fronteiras são controversas. A distinção entre gênero e estilo musical é similarmente controversa e redundante. Enquanto que gênero é geralmente considerado um conceito mais amplo e subjetivo em termos de conteúdo musical [Fabbri, 1999], não há um acordo geral sobre o que um estilo deve abranger ou se é aceitável o uso indiscriminado desses dois conceitos [Moore, 2001]. Muitas peças musicais são semelhantes, algumas mais semelhantes que outras. Além de servir como uma fonte infinita de discussão, tais semelhanças também são cruciais para o desenvolvimento de sistemas de recuperação da informação musical (do inglês, *Music Information Retrieval* - MIR) eficientes [Cilibrasi et al., 2004].

O crescimento constante dos dados musicais na Internet tem encorajado diversos pesquisadores a desenvolver ferramentas adequadas para a análise e a classificação destes dados. O objetivo principal de tais ferramentas é extrair a informação de forma compacta e representativa ao conteúdo dos bancos de dados. [Corrêa, 2012]. Este projeto de pesquisa consiste em desenvolver um algoritmo capaz de organizar automaticamente as partituras

monofônicas, sabendo que a classificação não supervisionada ou agrupamento de dados (do inglês, clustering), é relativamente pouco explorada para tarefas de classificação de gêneros musicais [Scaringella et al., 2006].

2. Revisão de Conceitos musicais

2.1. Entendendo o som

O som possui 4 propriedades:

- **Duração:** É o tempo que o som perdura ao ser executado.
- **Intensidade:** É a força do som, ou seja, força maior ou menor do som, a qual aumenta com a amplitude das vibrações.
- **Altura:** São as notas musicais (Dó, Ré, Mi, Fá, Sol, Lá, Si, Dó) em várias oitavas desde o grave até o agudo. O conjunto de notas existentes entre uma nota e a sua primeira repetição é chamado de oitava.
- **Timbre:** É através dessa propriedade que podemos distinguir a diferença entre sons da mesma nota (altura), duração e intensidade. Ex: a voz da criança, a voz da mulher e a voz do homem. Se pedirmos que essas pessoas cantem a mesma música separadamente, facilmente poderemos identificar quem está cantando, se uma criança, uma mulher ou um homem. Essa distinção entre as vozes é possível graças ao timbre.

Todo som produzido possui essas quatro propriedades, exceto alguns instrumentos de percussão, como o tambor, o bumbo, dentre outros.

2.2. Representação musical

Definição: Conjunto de sinais gráficos que permitem fixar em notação escrita, uma ideia musical com todos os seus detalhes: notas, pausas, intensidade, duração e timbre.

- **Nota:** Monossílabo que designa um som regular.
Sinal gráfico que representa a altura e duração dos sons musicais.[Med, 1996]
- **Pauta ou Pentagrama:** Utilizado para representar a notação musical.
A pauta ou pentagrama é composta por 5 linhas horizontais, paralelas e equidistantes entre si, formando 4 espaços ou entrelinhas, contadas de baixo para cima.
- **Clave:** é um sinal, colocado na extremidade esquerda da pauta, recebem essa denominação em virtude da posição onde são escritas representarem as notas sol, fá e dó, respectivamente, e com isto, fixa também o nome das outras notas, pela relação com a clave, as posições das claves e as notas estão exemplificadas na Figura 1. Há três tipos de clave: clave de sol, clave de fá e clave de dó.

2.3. MusicXML

MusicXML [Good, 2006] é um formato de arquivo de notação musical aberto, baseado em XML, utilizado para integrar partituras digitais entre aplicações. Foi inventado por Michael Good na empresa Recordare em 2000 e vem sendo desenvolvido com a colaboração de uma comunidade de centenas de músicos e desenvolvedores de software durante 13 anos. Disponível sob uma licença aberta, isenta de royalties, sendo vantajosa tanto para softwares open-source e softwares proprietários. Suportada por mais de 160 aplicações em todo o mundo.

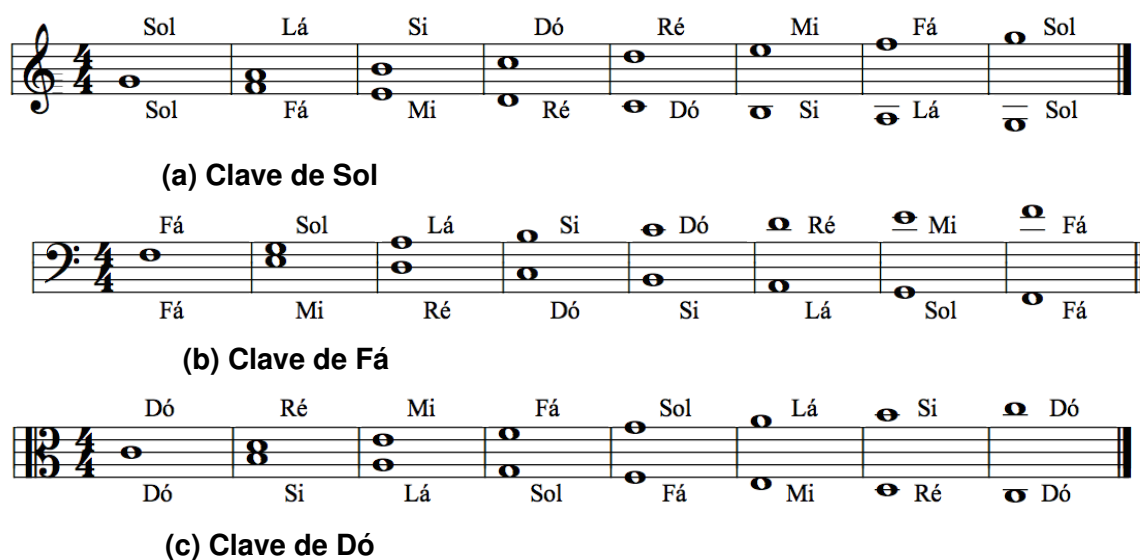


Figura 1: Exemplo das posição das claves e as notas representadas

3. Visão Geral

Como pode ser visto na Figura 2, tem-se como entrada o arquivo em formato musicXML, a partir deste, é feita a criação de um vetor de tóplas. O vetor de tóplas é utilizado como entrada do algoritmo de alinhamento proposto por [Mongeau and Sankoff, 1990]. Tem-se como resultado uma matriz de distância e a partir desta é gerado o dendrograma através do algoritmo de agrupamento hierárquico.



Figura 2: Visão geral do processo

Tabela 1: Mapeamento das notas musicais

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	1	2	3	4	5	6	7	8	9	10	11

3.1. Parser de musicXML

Para ser possível a comparação de sequências musicais a partir de partituras, foi desenvolvido um parser de musicXML, onde o software lê a partitura e gera um vetor de túplas, onde cada túpla tem a informação sobre a nota, sua oitava e a sua duração.

Formalmente, este procedimento é definido como dado uma música M vai ser composta de um vetor de túplas com 3 elementos $\langle A, B, C \rangle$ onde:

- A = Letra da nota musical ou @ caso seja uma pausa.
- B = Oitava da nota musical ou VAZIO \emptyset caso seja uma pausa.
- C = Duração de cada nota ou pausa. O elemento de duração é um inteiro que representa a duração de uma nota em termos de divisões por semínima.

Este procedimento esta exemplificado na Figura 3. Onde o primeiro item apresentado é parte de uma música em musicXML e depois é apresentado o resultado da geração do vetor de túplas, considerando apenas o trecho de código apresentado.

3.2. Transposição de Tons

Após a criação do vetor de túplas, é necessário transpor todas as músicas para o mesmo tom ser efetuada a comparação das mesmas. Esta etapa é importante, pois pode ocorrer de uma música ser similar a outra mas em uma tonalidade diferente, tendo a mesma melodia, porém com notas diferentes, e ao ser comparada, pode gerar resultados inconsistentes. Assim para efetuar a transposição em larga escala, foi criada a equação para transpor todas as partituras para o tom C (Dó), podendo assim gerar resultados consistentes. Para automatizar a transposição, cada nota foi associada a um conceito numérico, como apresentado na Tabela 1.

Então utilizando como exemplo a música infantil "Brilha, brilha estrelinha" no tom A (Lá), iremos transpo-la para o tom C (Dó), ou seja, iremos "descer" cinco tons e um semitom. Em notação escrita em valores tem-se a música no tom A (Lá) como: 9, 9, 4, 4, 6, 5, 4, 2, 2, 1, 0, 11, 11, 9. O tom de uma música, geralmente é a primeira nota tocada, portanto neste exemplo o tom é a nota A, ou escrita em valores 9.

A equação utilizada para a transposição é $NOTATRANSPOSTA = NOTA - TOM$. A nota A (Lá) escrita em valores equivale a 9, portanto para transpo-la utilizando a equação será $9 - 9 = 0$, onde 9 é a nota a ser transposta, ou seja, a nota A (Lá), o outro 9 é o tom da música, e o resultado 0 equivale a nota C (Dó).

Outro problema encontrado na transposição de tons foi quando uma nota está em uma oitava diferente da outra nota, não conseguindo transpor de forma correta algumas notas. Por exemplo a nota E (Mi), no tom A (Lá), transposta para o tom C (Dó) pela equação, seria a nota F (Fá), mas na realidade é a nota G (Sol). Para solucionar este problema, foi criada a equação $NOTAREAL = (OITAVA - 1) * 12 + NOTA$. Com isto encontra-se o real valor da nota a ser transposta, assim podendo transpor qualquer nota em qualquer oitava para o tom C (Dó), então transpondo a nota E (Mi) para o tom C (Dó), sabe-se que a tom está na quarta oitava, e a nota E (Mi) está uma oitava acima, ou seja, na quinta oitava. Então temos para a nota E (Mi) $(5 - 1) * 12 + 4 = 52$, e para a nota A (Lá) $(4 - 1) * 12 + 9 = 45$, então utilizando a equação da transposição temos $|52 - 45| = 7$, onde 7 equivale a nota G (Sol).

```

<note>
<pitch>
<step>D</step>
<octave>4</octave>
</pitch>
<notations>
<technical>
<fret>0</fret>
<string>4</string>
</technical>
</notations>
<voice>1</voice>
<duration>960</duration>
<type>quarter</type>
</note>
<note>
<pitch>
<step>C</step>
<octave>4</octave>
</pitch>
<notations>
<technical>
<fret>3</fret>
<string>5</string>
</technical>
</notations>
<voice>1</voice>
<duration>1920</duration>
<type>half</type>
</note>
</measure>
<measure number="5">
<note>
<rest/>
<voice>1</voice>
<duration>1920</duration>
<type>half</type>
</note>

```



...;D,4,960;C,4,1920;@,Ø,1920;...

Figura 3: Exemplo de como é criado vetor de túplas

3.3. O Algoritmo de Comparação

Na comparação de sequências musicais, existem alguns critérios a serem utilizados, estes são a dissimilaridade, as transformações típicas e os pesos.

A dissimilaridade entre duas sequências musicais, é dada a partir da menor quantidade de transformações típicas necessárias para uma sequência A transformar-se em uma sequência B. Foram utilizados três tipos de transformações: a Deleção, a Inserção e a Substituição. Para melhor exemplificar as transformações típicas e o conceito de dissimilaridade, segue o exemplo: Dado duas sequências A e B (Figura 5) para transformar A



(a) Música em tom A (Lá)



(b) Música em tom C (Dó)

Figura 4: Trecho da Música “Brilha, brilha estrelinha” em dois tons diferentes

em B será necessário efetuar a deleção da nota D (Ré) (Figura 6), a substituição da nota B (Si) pela nota C (Dó) (Figuras 7.a e 7b.) e a inserção da nota C# (Dó sustenido) (Figuras 7.c e 7.d).



Figura 5: Sequências A e B de notas musicais e representação das mesmas em partitura

Para transformar A em B, foram necessários 3 transformações, tendo uma dissimilaridade de valor 3.

O conceito de dissimilaridade entre duas notas, pode ser generalizado associando um peso a cada tipo de transformação, podendo assim, pré-definir valores relacionados para cada transformação (inserção, deleção, substituição) e os elementos envolvidos na transformação.

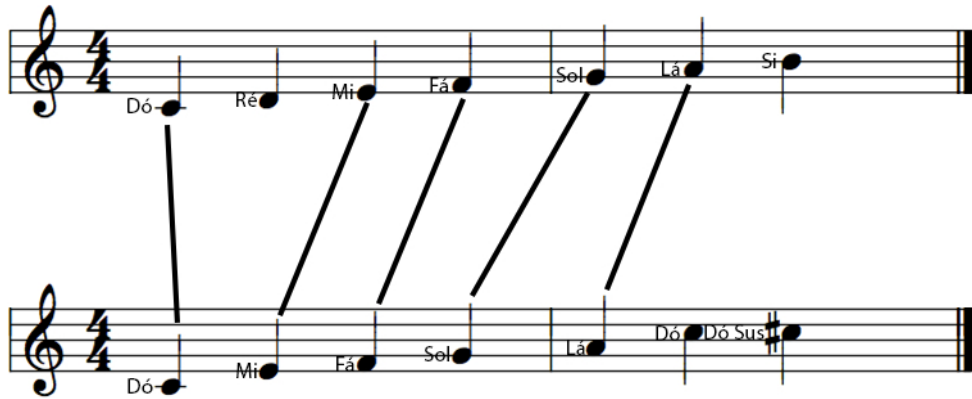
3.3.1. Pesos

A dissimilaridade vem da soma dos menores pesos de todas as possibilidades de transformação de A para B. Sendo $m_{i,j}$ a dissimilaridade entre as notas de A e B, pode ser visto na equação 1 os pesos associados a deleção, inserção e substituição [Mongeau and Sankoff, 1990].

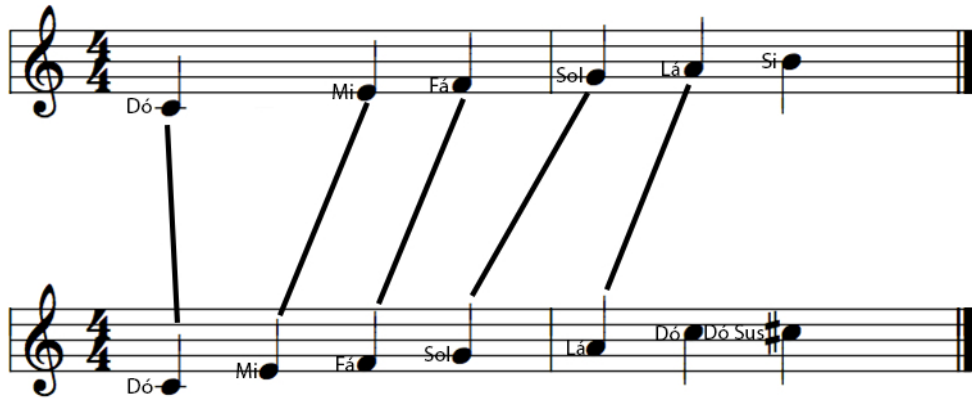
$$m_{i,j} = \min \begin{cases} m_{i-1,j} + w(a_i, \emptyset) & \text{(Deleção) ;} \\ m_{i,j-1} + w(\emptyset, b_j) & \text{(Inserção) ;} \\ m_{i-1,j-1} + w(a_i, b_j) & \text{(Substituição) ;} \end{cases} \quad (1)$$

O peso é definido pela soma de outros dois pesos, sua equação pode ser vista na equação 2.

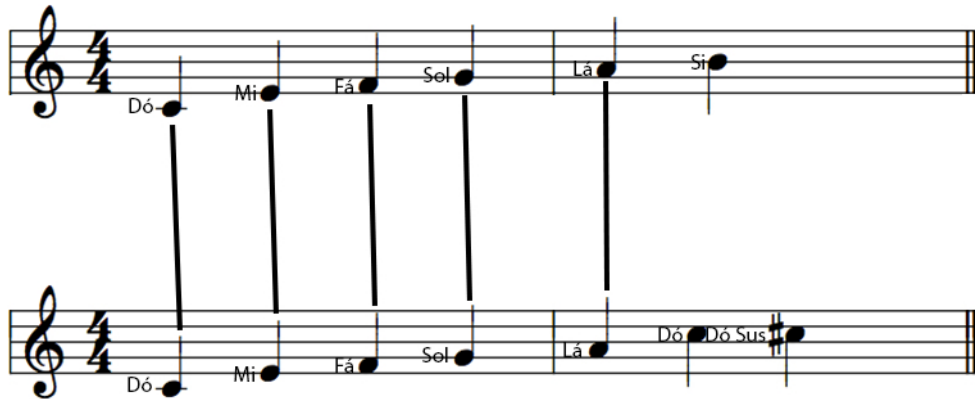
$$w(a_i, b_j) = w_{interval}(a_i, b_j) + k * w_{length}(a_i, b_j) \quad (2) \quad 171$$



(a) Exemplo da necessidade de deleção para o alinhamento.

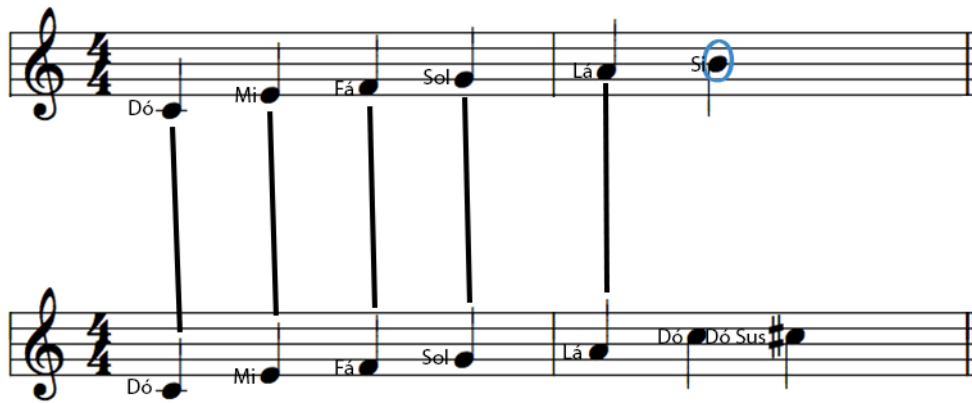


(b) Segunda nota do 1o compasso da 1a sequencia (Ré) removido.

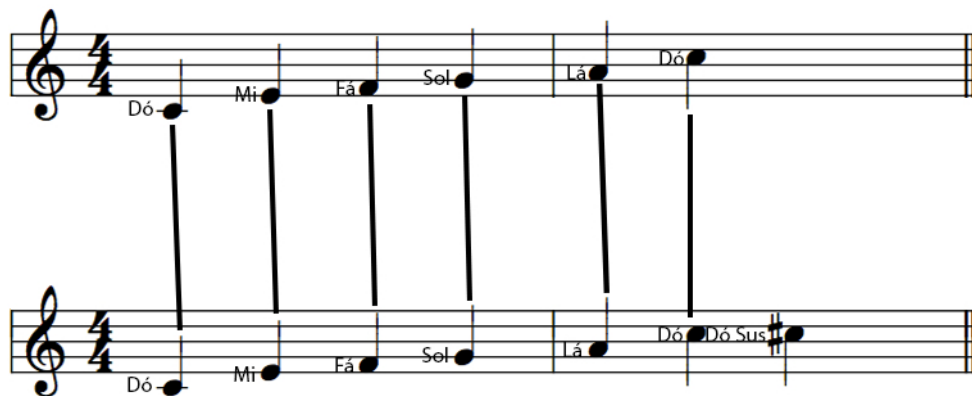


(c) Primeiro compasso de ambas as sequencias alinhados.

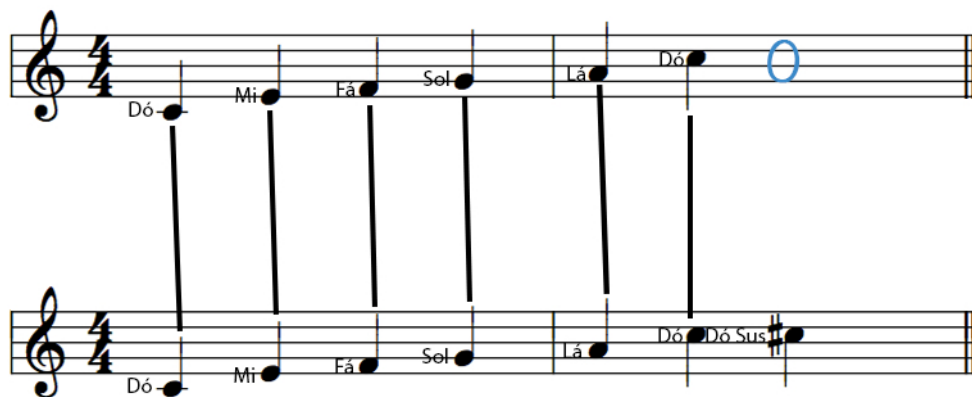
Figura 6: Exemplo de Deleção



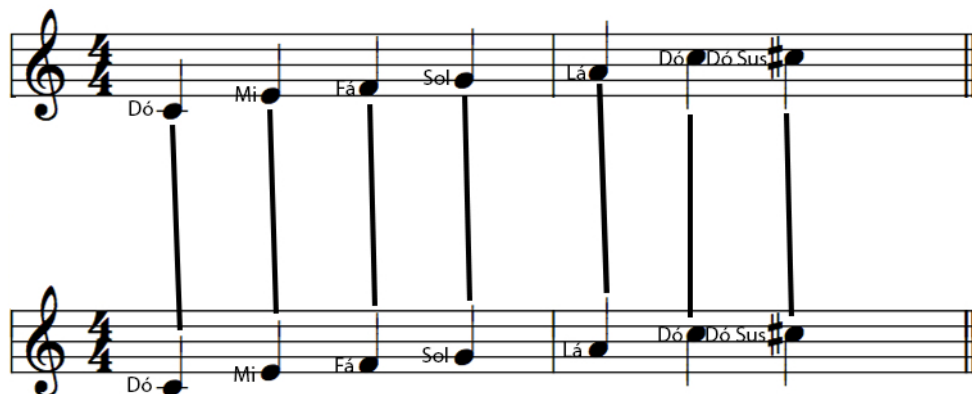
(a) Exemplo de Substituição: A 2a Nota do 2o Compasso (Si) vai ser substituída por um Dó pelo algoritmo de alinhamento.



(b) Resultado da substituição.



(c) Exemplo de Inserção: Para que as duas sequencias fiquem iguais é necessário incluir uma nota (Dó Sustenido) no 2o compasso.



(d) Resultado da inclusão.

Figura 7: Exemplo de Substituição e Inserção

Onde a_i e b_j são as notas das músicas A e B e k representa a contribuição relativa da diferença dos tamanhos pela diferença de notas e foi considerado com valor 3.

A função $w_{length}(a_i, b_j)$ é definida pela equação 3.

$$w_{length}(a_i, b_j) = ||a_i| - |b_j|| \quad (3)$$

A função $w_{interval}(a_i, b_j)$ é definida pelo grau de diferença entre a nota a_i e a nota b_j , onde cada grau é definido por:

$$\begin{aligned} deg(0) &= 0; \\ deg(4) &= 0.1; \\ deg(2) &= 0.2; \\ deg(5) &= 0.35; \\ deg(3) &= 0.5; \\ deg(6) &= 0.8; \\ deg(1) &= 0.9; \\ deg(> 6) &= 9999999999; \end{aligned}$$

4. Resultados e Discussões

Foram utilizadas vinte e cinco partituras no formato musicXML em cinco gêneros diferentes, apresentadas na Tabela 2.

Após a criação do vetor de túplas são feitas as transposições de todas as partituras do banco de dados, foram comparadas todas as músicas, par a par, utilizando o algoritmo proposto em [Mongeau and Sankoff, 1990]. Para efetuar a comparação, é necessário inicializar a matriz $m_{i,j}$, para $1 \leq i \leq m$, $1 \leq j \leq n$, definindo os pesos associados a deleção e inserção entre duas músicas a serem comparadas.

$$\begin{aligned} m_{i,0} &= m_{i-1,0} + w(a_i, \emptyset) \quad , i \geq 1; \\ m_{0,j} &= m_{0,j-1} + w(\emptyset, b_j) \quad , j \geq 1. \end{aligned}$$

Então a partir da aplicação da equação 1, é gerado um escore da comparação entre duas músicas. Após efetuar a comparação par a par entre todas as músicas, foi construído uma matriz de distância, e utilizando o método de aglomeração hierárquica, construiu-se um dendrograma. Uma matriz de distâncias é onde cada linha explicita a hierarquia de proximidade de cada observação em relação às restantes, para cada observação, a proximidade com as outras será maior, quanto maior for a similaridade que lhes corresponde.

O algoritmo de aglomeração hierárquica progressivamente agrupa as N amostras em C grupos, de acordo com um parâmetro definido, geralmente associado a uma medida de distância ou similaridade entre os vetores de atributos das amostras. No primeiro passo ($k = 1$), as N amostras são atribuídas como N grupos. No passo seguinte ($k = 2$), ocorre uma partição de $N - 1$ grupos, em seguida uma partição de $N - 2$ grupos e assim por diante. No último passo ($k = N$), todas as amostras fazem parte de um único grupo. Portanto, esta sequência progressivamente agrupa amostras que são mais similares entre si antes de considerar amostras menos similares. É possível dizer que, no k -ésimo passo, $C = N - k + 1$ (C é o número de grupos) [Corrêa, 2012].

O dendrograma é a representação gráfica do processo de constituição dos clusters, onde permite identificar os clusters agrupados ao longo de todo o processo e observar o incremento nos valores da distância entre os clusters.

Tabela 2: Código, Gênero, Artista e Título das partituras selecionadas

Código	Gênero	Artista	Título
BOL1755	Bolero	Luis Miguel	Inolvidable
BOL1758	Bolero	Francisco Xavier	Recuerdos de Ipacará
BOL1764.2	Bolero	Linda Ronstadt	Perfídia
BOL1770.2	Bolero	Francisco Xavier	El reloj
BOL1770	Bolero	Francisco Xavier	El reloj
BOS0001	Bossa Nova	Chico Buarque	Bye, Bye brasil
BOS0002	Bossa Nova	Tom Jobim	Águas de março
BOS0003	Bossa Nova	Noel Rosa	A.B. surdo
BOS0004	Bossa Nova	Chico Buarque	A Rita
BOS0005	Bossa Nova	Tom Jobim	Chega de saudade
PAG2961	Pagode	Arlindo Cruz	Alem do meu querer
PAG2964	Pagode	Arlindo Cruz	Brasil Moleque
PAG2970	Pagode	Arlindo Cruz	Será que é amor
PAG2972	Pagode	Arlindo Cruz	Um sim para que te gosta
PAG2973	Pagode	Arlindo Cruz	Malandro sou eu
SER2614	Sertanejo	Rio Negro & Solimões	Tô Mal
SER2635	Sertanejo	Rio Negro & Solimões	Bate o pé
SER2644	Sertanejo	Rio Negro & Solimões	Saudade pulou no peito
SER2696	Sertanejo	Bruno & Marrone	Vai dar namoro
SER2822	Sertanejo	Leandro & Leonardo	Entre tapas e beijos
TAN6.2	Tango	Horacio Salgán	Los Mareados
TAN6.3	Tango	Horacio Salgán	Los Mareados
TAN48.3	Tango	Carlos Cardel	Caminito
TAN56.5	Tango	Carlos Gardel	Adios Muchachos
TAN81.5	Tango	Astor Piazzolla	Libertango

Analisando a figura 8, pode-se perceber que algumas músicas tal como as músicas do gênero sertanejo e pagode, foram agrupadas em uma mesma aglomeração. Percebe-se também que as variações da música TAN6 (TAN6.2 e TAN6.3) foram agrupadas unicamente em uma mesma aglomeração.

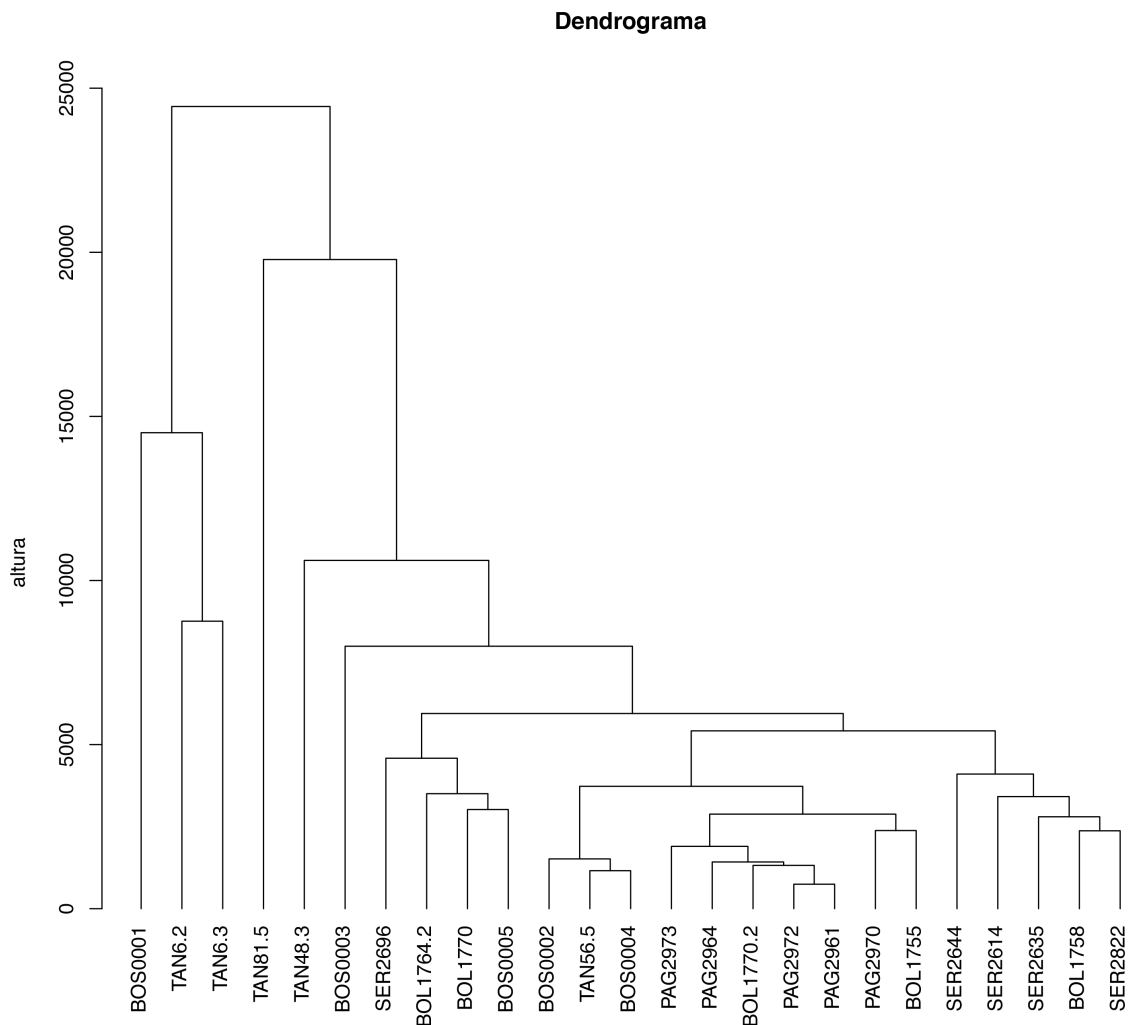


Figura 8: Dendrograma: Os prefixos BOS, TAN, SER, PAG, e BOL são, respectivamente, bossa nova, tango, sertanejo, pagode e bolero

5. Conclusões

Neste trabalho foi proposta uma abordagem para a organização automática de partituras utilizando um algoritmo de comparação de seqüências musicais e a técnica de agrupamento hierárquico. Foram realizados experimentos utilizando uma base com vinte e cinco partituras em formato musicXML de cinco gêneros musicais diferentes (bolero, bossa nova, pagode, sertanejo e tango). O método proposto se mostrou capaz de agrupar corretamente as partituras. Como trabalho futuro pretendemos fazer outros experimentos utilizando uma base de dados com um maior numero de partituras.

Referências

- Cilibrasi, R., Vitányi, P., and de Wolf, R. (2004). Algorithmic clustering of music based on string compression. *Computer Music Journal*, pages 49–67.
- Corrêa, D. C. (2012). *Inteligência artificial aplicada à análise de gêneros musicais*. PhD thesis, Universidade de São Paulo.

- Fabbri, F. (1999). Browsing music spaces: categories and the musical mind. *International Association For The Study of Popular Music*, pages 1–14.
- Good, M. (2006). Lessons from the adoption of musicxml as an interchange standard. In *Proceedings of the XML Conference*.
- Holt, F. (2007). *Genre in popular music*. University of Chicago Press.
- Huron, D. (2000). Perceptual and cognitive applications in music information retrieval. *International Society for Music Information Retrieval*, pages 1–6.
- Med, B. (1996). *Teoria da música*. Musimed.
- Miranda, E. R. (2002). *Composing music with computers*. Focal Press.
- Mongeau, M. and Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175.
- Moore, A. F. (2001). Categorical conventions in music discourse: style and genre. *Music & Letters*, 82:432–442.
- Priolli, M. L. d. M. (2007). *Princípios básicos da música para a juventude*, volume 1. Casa Oliveira de Músicas LTDA.
- Scaringella, N., Zoia, G., and Mlynek, D. (2006). Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23:133–141.