Frequency tree for analysing Estonian folk song regilaul tunes

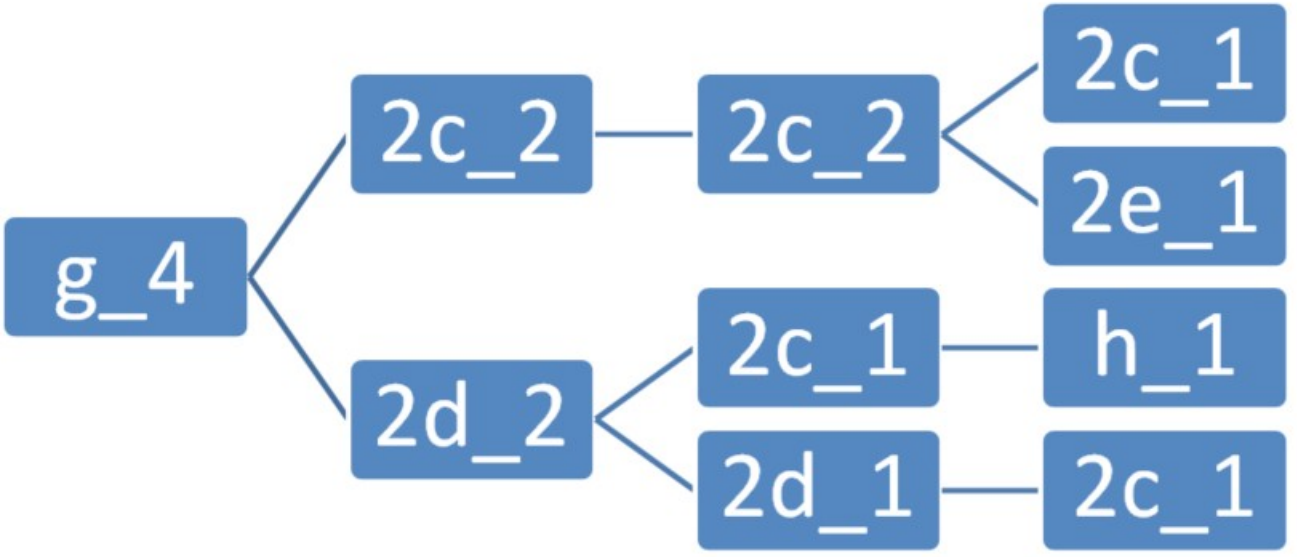
Jaagup Kippar

Similarity finding

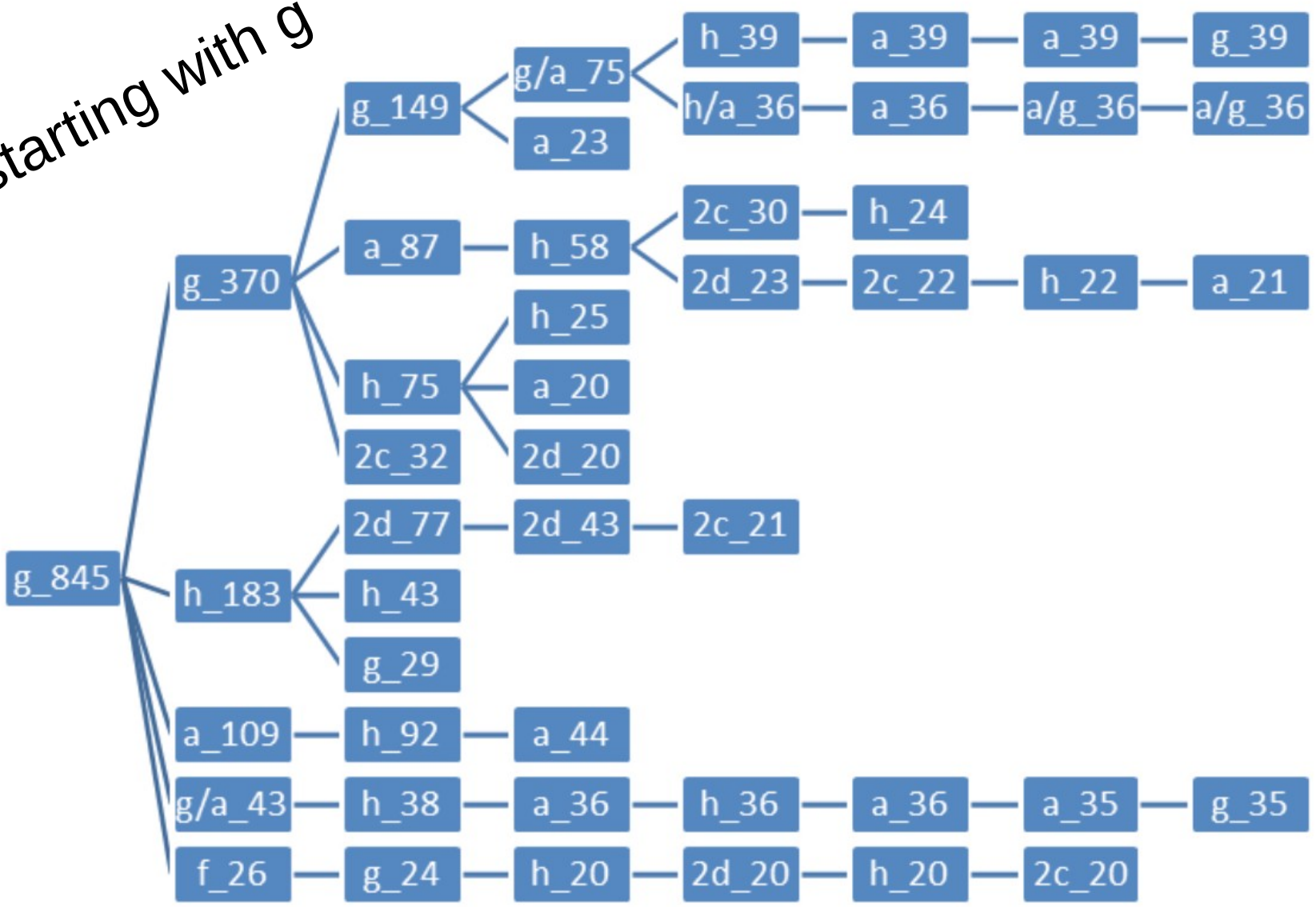
- We don’t have „one and good” way to say, that tunes are similar
- We can group tunes by some properties like starting note, starting interval, starting notes bi- tri- or tetragram
- One universal method for describing and then comparing sequential data is frequency tree

Frequency tree example

g	2c	2c	2c
g	2c	2c	2e
g	2d	2c	h
g	2d	2d	2c



Tunes starting with g

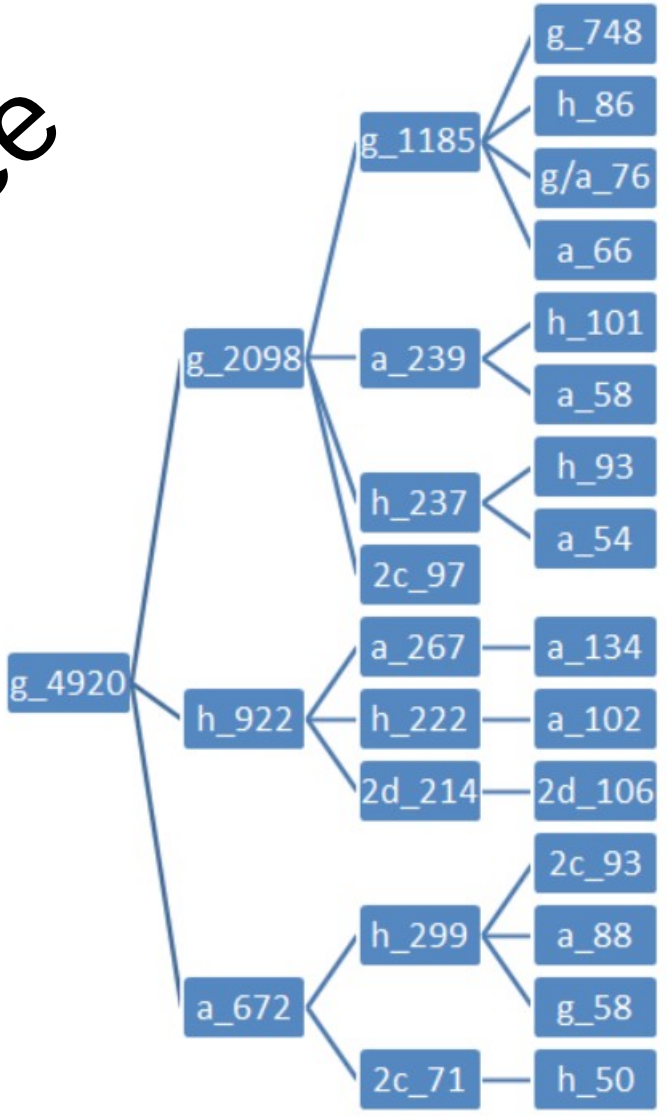


Sequences by window

```
g a h 2c a g h a


g a h 2c
a h 2c a
h 2c a g
2c a g h
a g h a
```

Window sequences tree

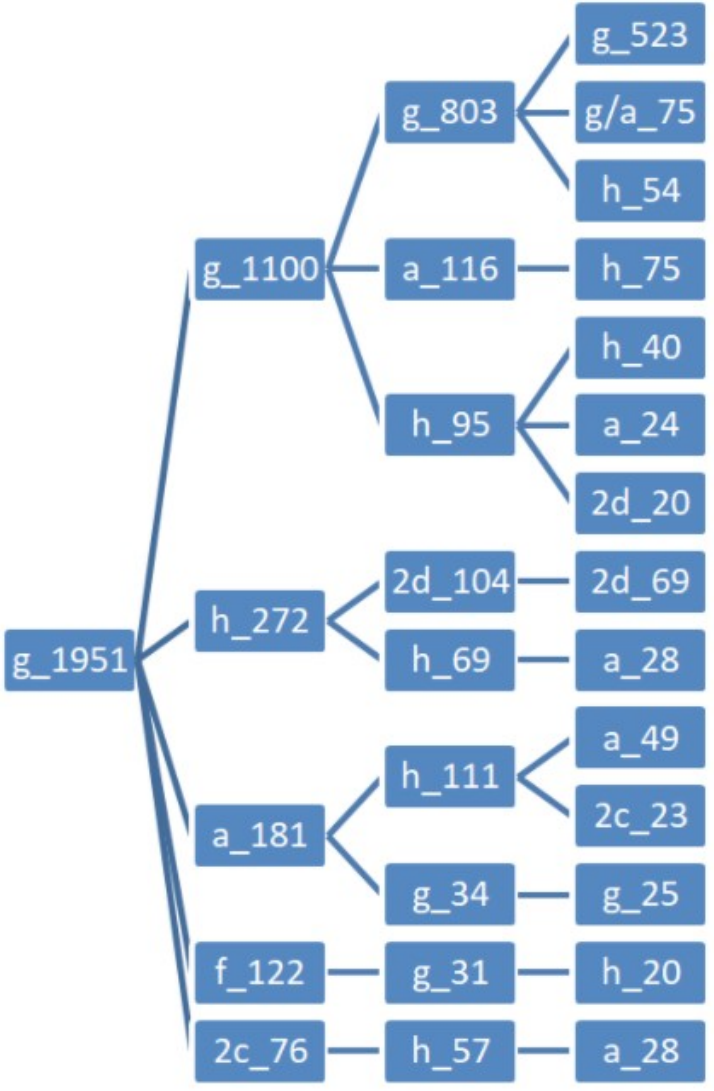


Rhythmic part sequences

g a h 2c a g h a g a h 2c a g g g

g a h 2c
a g h a
g a h 2c
a g g g

Rhythmic part sequences



CSV-file with tunes

```
IDviis,FKey,VNr,Nr,voti,alter,tugiheli,korgus,tempo,takt,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P
11,P12,P13,P14,P15,P16,markusi,viis
68,245,2,,&,#f,g,b,,,2d,h,h,g,h,a,a,f[g],2c,a,a,f,a,g,g,g,,alg
69,246,1,,&,,g,,,,2d,2d,2c,h,2d,h,h,a,2c,h,a,g,h,g,g,g,,
70,247,1,,&,#f,g,g1=Fis,,,2d,2d,2d,2c,h,a,a,h,2c,h,2c,a,a,g,g,g,,
71,4492,1,,&,#f,g,h,84,24,g,d,2c,2c,h,a,g,e,e,d,d[h][a],a[f],a,f [a][a/f],g,g,,
72,4563,1,,&,#f,g,,,,h,2d,h,2c,h,g,2d,h,a,g,2c,h,a[h],g,g,g,,4563
74,248,1,,&,#f,g,,,,2d,h,g,g,h,a,f,a,2c,a,f,f,g,g,g,h,,
75,249,1,,&,#f,g,,,,2d,h,g,2d,h,g,g,g,2d,h,f,2d,h,g,g,g,,
76,250,1,,&,#f,g,,,,2d,h,g,2d,2c,a,f,a,2c,a,f,2d,h,g,g,g,,
77,251,1,,&,bh,f;a,,,,2c,a,2d,2c,h,g,g,a,2c,h,h,g,h,a,a,a,,
78,252,1,,&,#f;#c;#g,#c,,,,2e,2d,2c,2d,2e,2c,2c,2c,2d,2c,h,h,2e,2d,2c,2c,,252
79,253,1,,&,#f;#c;#g,a;#c,,,,2e,2d,2c,2d,2e,2c,2c,2c,2d,2c,h,h,2e,2d,2c,2c,,253
80,254,1,,&,#f;#c;#g,a;#c,,,,2e,2d,2c,2d,2e,2c,2c,2c,2d,2c,h,h,2e,2d,2c,2c,,254
81,255,1,,&,#f,g,,,,2d,h,2d,2e,2d,2c,a,2d,h,g,a,2c,h,g,g,g,,
```

Generating tree

```
root={"amount":0}
f=open("regiviisid.csv")
f.readline()
for line in f:
 pointer=root
 pointer["amount"]+=1
 for note in line.strip().split(",")[10:13]:
  if note in pointer:
   pointer[note]["amount"]+=1
  else:
   pointer[note]={"amount":1}
  pointer=pointer[note]
```

Displaying tree

```
def display(node, level):
  for key in node:
   if key!="amount":
    print(" "*level+key+"_"+str(node[key]["amount"]))
    display(node[key], level+1)

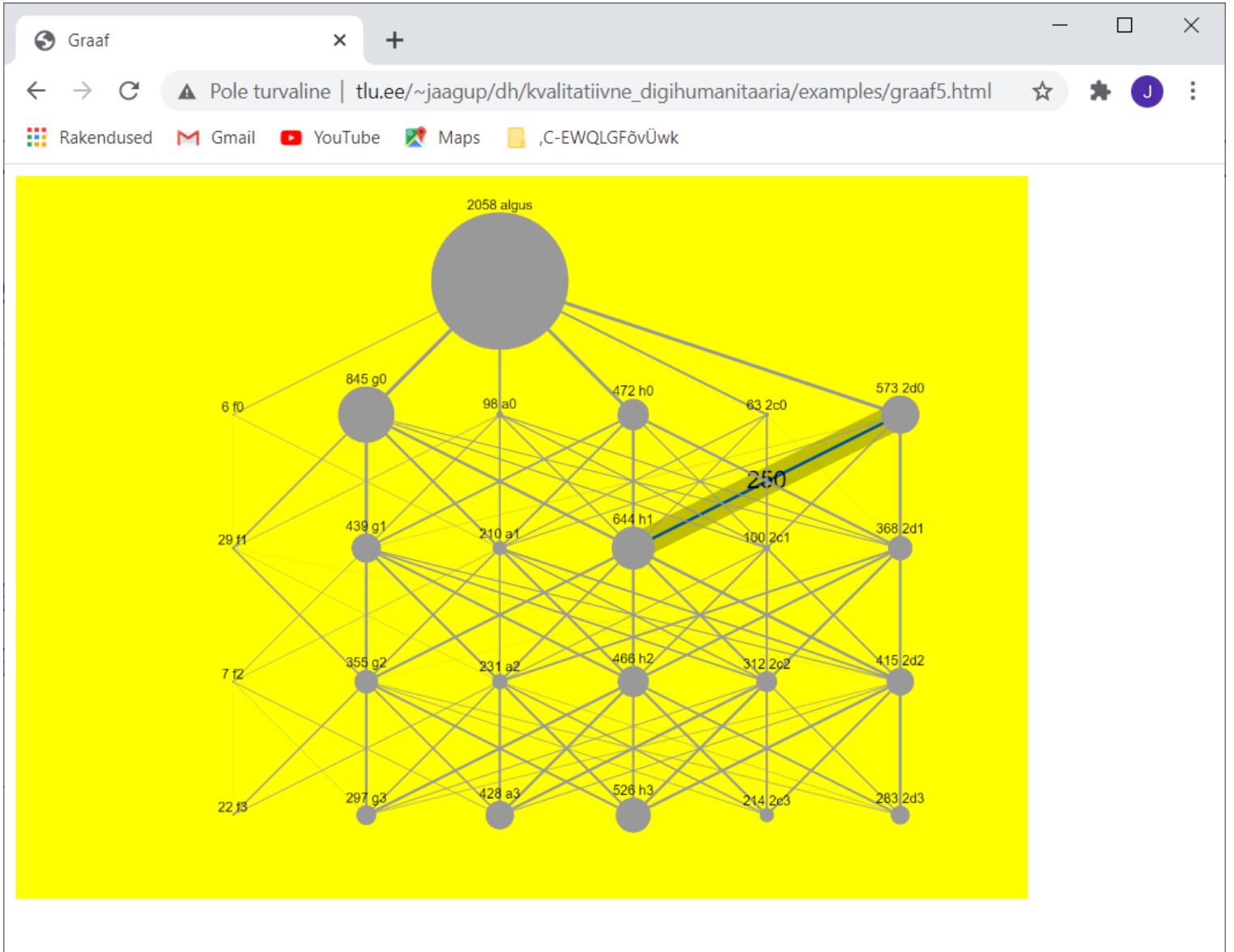
display(root, 0);
```

Frequency tree with map

- 2d 151
  - h 78
    - h 50
      - g 50
  - 2d 31
    - h 29
      - g 29
  - 2d[2c] 26
    - h 26
      - g 26
- g 142
  - g 60
  - h 39
- h 102
  - h 64
    - g 50
      - g 50
- 2c 74



Tune graph


Graaf
Pole turvaline | tlu.ee/~jaagup/dh/kvalitatiivne_digihumanitaaria/examples/graaf5.html
Rakendused
Gmail
YouTube
Maps
,C-EWQLGFõvÜwk
2058 algus
845 g0
98 a0
472 h0
63 2c0
573 2d0
6 f0
250
439 g1
210 a1
644 h1
100 2c1
368 2d1
29 f1
355 g2
231 a2
466 h2
312 2c2
415 2d2
7 f2
297 g3
428 a3
526 h3
214 2c3
283 2d3
22 f3

Frequency tree
for analysing Estonian folk song regilaul tunes

Jaagup Kippar