

# Szakdolgozat

*Balázs Ádám*

*Kuk József*

*Debrecen*

*2010*

**Debreceni Egyetem**

**Informatika Kar**

# **EKG jelek feldolgozása (.NET)**

Témavezető:

**Dr. Juhász István**

Egyetemi adjunktus

Készítette:

**Balázs Ádám**

Programtervező Informatikus BSc.

**Kuk József**

Programtervező Informatikus BSc.

*Debrecen*

*2010*

## Tartalomjegyzék

1.	Bevezetés .....	5
2.	Szószedet .....	7
3.	Az Optics algoritmus .....	8
3.1	Klaszterezés .....	8
3.2	Az Optics .....	8
3.2.1	Az algoritmus paramétereit.....	9
3.2.2	Magpontok.....	9
3.2.3	Az Optics algoritmus .....	9
3.2.4	Az algoritmus kimenete.....	10
3.2.5	Az algoritmus megjelenése a programban.....	10
4.	Fejlesztői dokumentáció .....	12
4.1	Adattároló és feldolgozó osztályok.....	13
4.2	Megjelenítést végző osztályok.....	15
4.3	Az osztályok rövid leírása.....	18
4.3.1	Labtech.EcgViews névtér.....	18
4.3.2	Labtech.EcgViews.Ecg névtér.....	18
4.3.3	Labtech.EcgViews.Ecg.Filters névtér .....	19
4.3.4	Labtech.EcgViews.EcgDraw névtér.....	19
4.3.5	Labtech.EcgViews.UI névtér.....	19
5.	Felhasználói Dokumentáció .....	20
5.1	Főablak.....	20
5.1.1	A menüsor elemei.....	21
5.1.2	A program főablaka a fájlfeldolgozás után.....	23
5.1.3	Az 5.7. ábra jelmagyarázata: .....	24
5.2	A program paramétereinek beállítása .....	25

5.2.1	Paraméterek megadása FLT állomány használata esetén.....	26
5.2.2	Paraméterek megadása ANT állomány használata esetén.....	28
5.3	A program használata .....	29
5.3.1	A program használatának megkezdése, a felvétel kiválasztása.....	29
5.3.2	A főablakon végezhető műveletek .....	29
5.3.3	A kézi klaszterezés folyamata .....	35
5.3.4	Az automatikus klaszterezés folyamata.....	39
5.3.5	Klaszterezési módszerek keverése.....	40
6.	Összefoglalás .....	42
7.	Köszönetnyilvánítás .....	43
8.	Irodalomjegyzék .....	44

# 1. Bevezetés

A Labtech Kft. 2008-ban kereste meg a Debreceni Egyetemet egy, már létező kardiológiai termékük továbbfejlesztésének céljából. Az Egyetemen munkacsoport alakult a fejlesztések megvalósítására Dr. Juhász István Tanár Úr vezetésével. A csoport további tagjai Vágner Anikó Tanárnő, Balázs Ádám valamint Kuk József hallgatók. A Labtech Kft. részéről, a projekt vezetését, a cég és az egyetem közötti kapcsolattartást Farkas László végezte.

Jelen dolgozatban - melyet Balázs Ádám és Kuk József, a projektben résztvevő két hallgató készített - bemutatjuk az elkészült alkalmazást mind felhasználói, mind programozói szemszögből. Mielőtt elkezdenénk a téma részletes tárgyalását, szeretnénk néhány szót szólni arról, hogy mi is a projekt célja, miért volt szükség a jelen szakdolgozat témájául szolgáló szoftver elkészítésére.

Bizonyos kardiológiai megbetegedések kivizsgálása érdekében a páciens EKG jelének hosszú ideig történő megfigyelése és annak kivizsgálása szükséges, miközben napi rutinjának megfelelően viselkedik. Ilyen esetekben egy kompakt készülék (úgynevezett Holter készülék) kerül elhelyezésre a beteg testén, mely általában 24 órán keresztül rögzíti a digitalizálandó EKG jelet. A rögzített felvétel teljes, manuális kielemezése azonban annak hossza miatt rengeteg idejébe kerülne az orvosnak, ezért a számítógépes elemzés szinte elkerülhetetlen. Ezen elemzés egyik fázisának, az úgynevezett *template készítésnek* továbbfejlesztésére kért fel minket a Labtech Kft.

A template készítés célja röviden összefoglalva az, hogy az egyes szívütéseket, azok formáját figyelembe véve egy csoportba, klaszterbe rendeljük a hasonlókat, majd előállítsunk egy, a klaszterben található szívütések jellemző formáját meghatározó görbét, a template-et. Az elemzés során létrehozott klaszterek által reprezentált szívütés-csoportok között az elemző sokkal könnyebben eligazodik, könnyebben kiválasztja a rendellenes ütések, mintha azt a teljes felvételen tenné. Az egyes klaszterekben lévő szívütések darabszámát, a klaszterhez tartozó template-et, valamint magukat a szívütéseket elemezve az orvos diagnosztikailag hasznos következtetéseket tud levonni a beteg állapotára vonatkozóan, melynek tárgyalása kívül esik jelen szakdolgozat témáján.

Az alkalmazás fejlesztése során a programozási munkákat Balázs Ádám és Kuk József hallgatók végezték.

Balázs Ádám volt a felelős az alkalmazás felhasználói felületéért, a megjelenítést végző osztályokért, valamint a kézi klaszterezés megvalósításáért. [10]

Kuk József feladata volt a felvételek-állományok feldolgozása [6], a szélsőérték-keresés [7] [8], valamint az automatikus klaszterezés [9] megvalósítása.

A Fejlesztői Dokumentáció osztályleírásainál (4.3 fejezet) tételesen fel van tüntetve az osztályok neve mellett, hogy melyik hallgató dolgozott rajta.

## 2. Szószedet

Kifejezés	Leírás
Holter készülék	A beteg EKG jelét figyelő és rögzítő berendezés, mely kimenete az EKG felvétel.
EKG felvétel	Digitalizált EKG jel, mely a feldolgozás alapjául szolgál.
annotáció	Az annotáció meghatározza, hogy az EKG felvétel mely időpillanataiban dobbant a páciens szíve. Az annotáció elkészítése egy külső szoftver feladata.
QRS	Az alkalmazás vonatkozásában az EKG jelnek egy olyan része, mely egy-egy szívütéshez tartozó annotáció egy bizonyos környezetében van. A dolgozatunkban QRS alatt a fenti fogalmat használjuk. Orvosi értelmezése eltérő.
csatorna	Egy EKG felvétel több csatornát is tartalmazhat, mely a szív eltérő irányokból történő megfigyeléséből előálló EKG jel.
klaszter	Jelen kontextusban QRS-ek egy csoportja.
klaszterezés	Az a folyamat, mely során az EKG felvétel QRS-eit klaszterekbe soroljuk. [3] [4]
Optics	A klaszterezés alapjául szolgáló algoritmus. Résztelesen a <u>3. fejezetben</u> kerül tárgyalásra.
FLT állomány	Egy külső szoftver által, az EKG felvételtől előállított jel, mely a szélsőérték-keresés alapjául szolgál.
szélsőérték pontok	Jelen kontextusban egy QRS-hez tartozó, a szélsőérték-keresés során előállított pontpár, mely a klaszterezés alapjául szolgálhat.
szélsőérték-keresés	Az FLT állományon elvégzett keresési folyamat, mely során minden egyes QRS-hez előállnak a hozzá tartozó szélsőérték pontok.
ANT állomány	Az EKG felvételtől tartozó Wavelet pontokat tartalmazó állomány.
wavelet pont	Wavelet transzformáció során előállított, egy-egy QRS-hez tartozó pont, mely a klaszterezés alapjául szolgálhat.
template	A klaszterben található QRS-eket jellemző görbe.

### 3. Az Optics algoritmus

Az automatikus klaszterezést az Optics algoritmus egy, a feladathoz illeszkedő implementációjával végezzük el a szoftverben. Ebben a fejezetben az algoritmus alapjait szeretnénk bemutatni, inkább alkalmazási megvilágításban, mintsem matematikaiban.

#### 3.1 Klaszterezés

A klaszterezés egy adathalmaz pontjainak, rekordjainak hasonlóság alapján való csoportosítását jelenti [3]. A klaszterezés, illetve osztályozás során az alappontokat diszjunkt csoportokba (klaszterekbe) soroljuk, azaz partícionáljuk az adathalmaz elemeit.

A klaszterezés célja, hogy az elemeknek egy olyan partícióját adjuk meg, amelyben a közös osztályba kerülő elempárok lényegesen hasonlóbba egymáshoz, mint azok a pontpárok, amelyek két különböző osztályba sorolódtak.

A klaszterezés során a megfelelő csoportok kialakítása nem egyértelmű feladat, hiszen a különböző adatok eltérő jelentése és felhasználása miatt más és más szempontok szerint kell csoportosítanunk. További nehézséget jelent, hogy egy  $n$  darab adatpontot tartalmazó adathalmaznak Bell-számnyi  $O(e^{n \lg n})$  darab klaszterezése lehet, vagyis az adatpontok számában exponenciálisnál nagyobb méretű keresési térben kívánunk egy optimális klaszterezést megtalálni. Ennek kiválasztásában a [4] volt segítségünkre.

#### 3.2 Az Optics

Több klaszterezési módszert megvizsgálva, jelen probléma megoldására az Optics [2] algoritmus használata mellett döntöttünk. Az algoritmus tulajdonképpen csak előkészíti a klaszterezést, rendezi az adathalmaz pontjait, a kapott rendezést megjelenítve, illetve automatikusan elemezve, a klaszter-struktúra azonosítható.

### 3.2.1 Az algoritmus paraméterei

Az algoritmus két paraméterrel rendelkezik:

$\varepsilon$  : Generáló távolság

$\mu$  : Sűrűségi korlát

### 3.2.2 Magpontok

Egy pont  $\varepsilon$ -sugarú környezete:  $N_\varepsilon(p) = \{q \in D \mid d(p, q) \leq \varepsilon\}$

Egy  $p \in P$  pont *magpont*, ha  $|N_\varepsilon(p)| \geq \mu$

Egy  $p \in P$  pont *határpont*, ha  $|N_\varepsilon(p)| < \mu$

A  $p \in P$  pont *közvetlenül elérhető* a  $q \in P$  pontból, ha

- 1)  $p \in N_\varepsilon(q)$  és
- 2)  $q$  magpont.

Egy  $p \in P$  pont *k-távolsága* az a  $d(q, p)$  távolság, melyre igaz, hogy

- 1) legalább  $k$  olyan  $r \in P \setminus \{p\}$  pont van, melyre  $d(p, r) \leq d(p, q)$
- 2) legfeljebb  $k - 1$  olyan  $r \in P \setminus \{p\}$  pont van, melyre  $d(p, r) < d(p, q)$

A  $p \in P$  *magpont magtávolsága* a  $\mu$  távolsága  $p$ -nek.

A  $p \in P$  pont *elérhető-távolsága*  $o \in P$  magponttól:  $\max(\text{magtávolság}(o), d(p, o))$

### 3.2.3 Az Optics algoritmusa

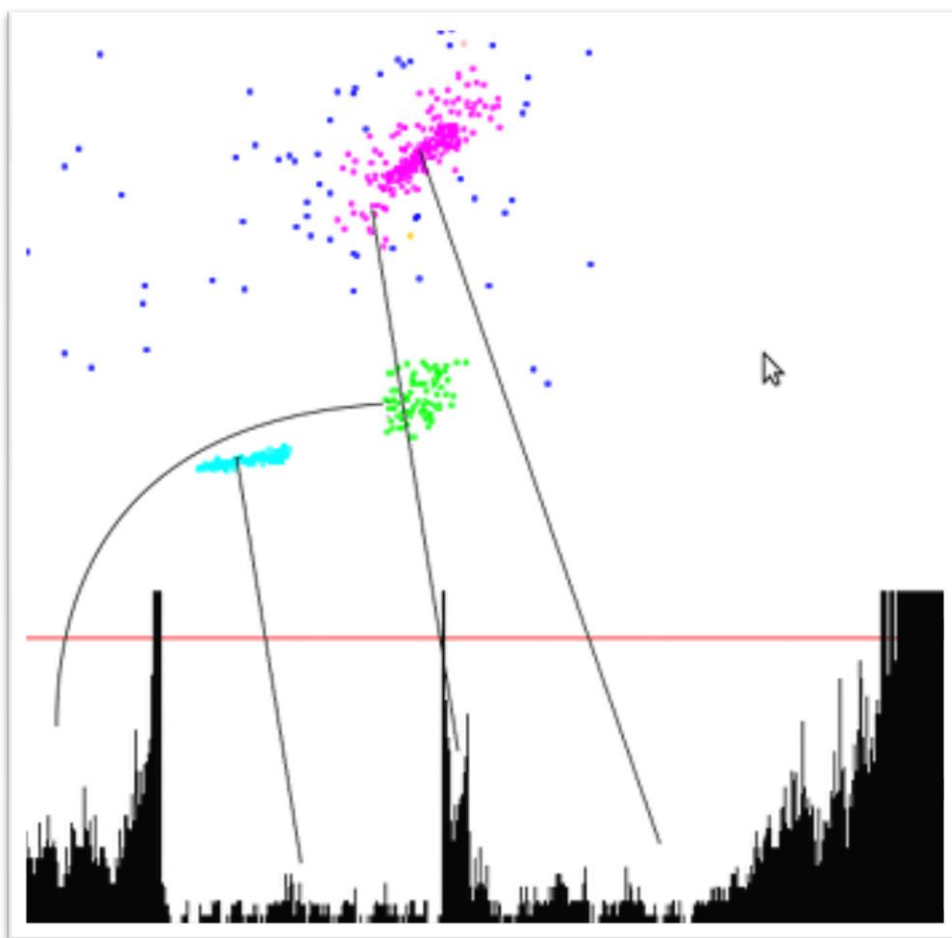
1. Vesz egy elemet a még nem vizsgált pontok halmazából.
2. Ha az elem nem magpont, akkor berakja az output halmazba.
3. Ha az elem magpont, akkor egy bővítési halmazba rakja a pont szomszédjait, magát a pontot pedig az output halmazba.
4. A bővítési halmaz minden elemére meghatározza az output halmaztól mért legkisebb távolságot. E távolság alapján rendezzi a bővítési halmazt.
5. Kiválasztja a bővítési halmaz legelső elemét, berakja az output halmazba, majd a szomszédjaival bővíti a bővítési halmazt.
6. Visszalép 1-re, ha a bővítési halmaz üres, egyébként visszalép 4-re.

### 3.2.4 Az algoritmus kimenete

Az algoritmus kimenete

- a  $P$ -beli alappontok lineáris rendezése;
- minden adatponthoz annak magtávolsága; valamint
- a pont környezetét jellemző  $r(p)$  érték, amely egy alkalmasan választott másik ponttól tekintett elérhetőségi távolság lesz.

Egy ilyen kimenetet szemléltet a következő, 3.1. ábra.



3.1. ábra: Az Optics algoritmus kimenete

### 3.2.5 Az algoritmus megjelenése a programban

Az algoritmus ugyan hatékony működésre képes és alkalmas a különböző formájú szívtűések szétválogatására, nagy erőforrásigénnyel rendelkezik. [5] Tekintve, hogy egy 24 órás felvétel

esetében akár több mint 100 ezer szívütés feldolgozása szükséges, ez igen nagy problémát jelent. Az algoritmus lefutási ideje 100 ezer szívütés esetén, még annak optimalizálása után is több mint másfél óra volt, mely nem elfogadható.

A probléma megoldására dolgoztuk ki a *rács technikát*, mely során, a síkon egy bizonyos kis távolságon belül található pontokat egy csoportba foglaljuk és az Optics algoritmus szempontjából egy pontnak tekintjük. Figyelembe véve, hogy az egyes szívütésekhez tartozó síkbeli pontok sok esetben ugyanoda, vagy egymáshoz nagyon közel esnek, ez a transzformáció nagyon minimális információveszteség mellett, jelentősen meggyorsítja az algoritmus lefutását.

Az algoritmus kiválasztásában és kidolgozásában az [1] volt segítségünkre.

## 4. Fejlesztői dokumentáció

Az alkalmazást a Visual Studio 2005 fejlesztőeszközzel, C# nyelven, .NET 2.0 környezet alá készítettük el.

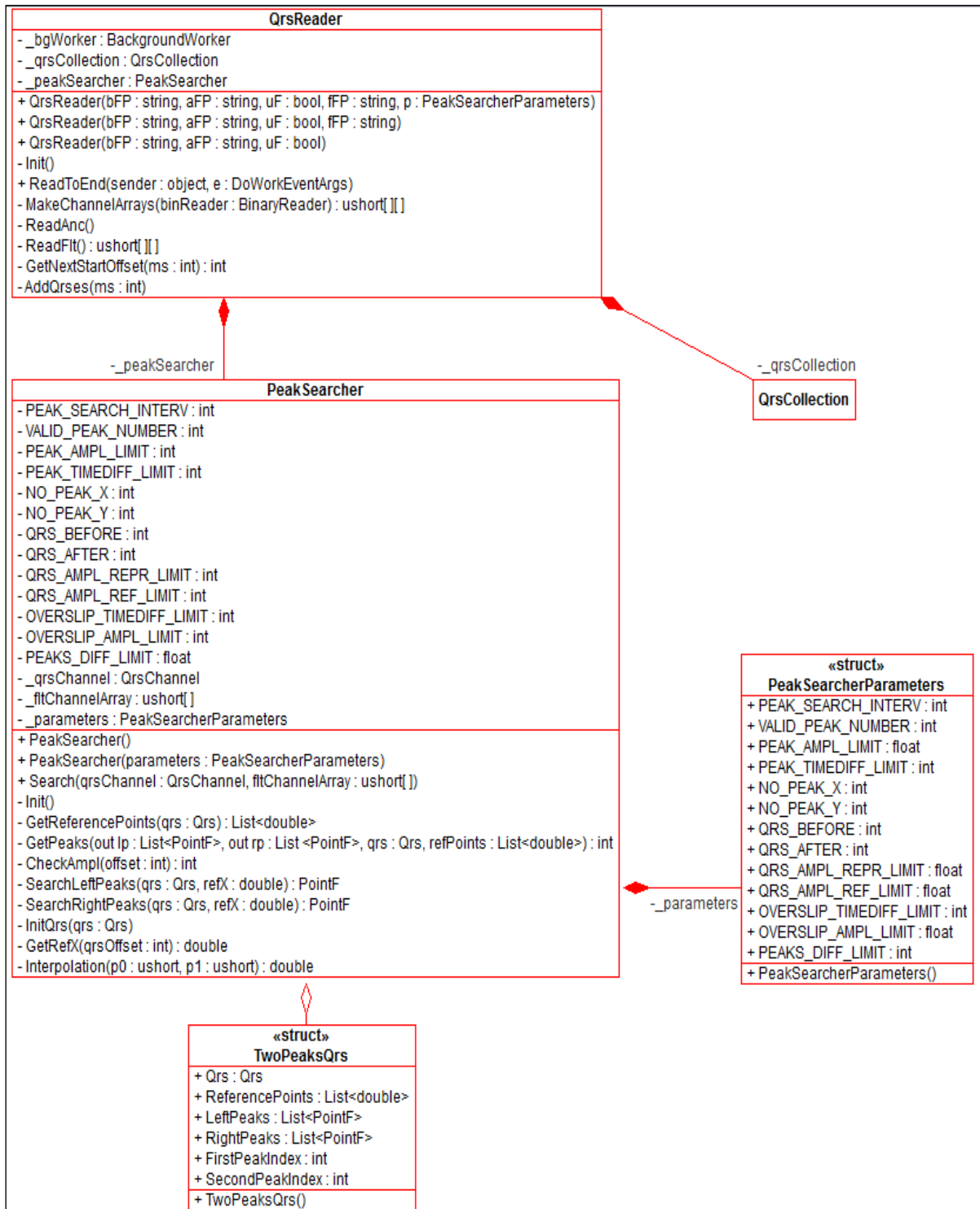
A program fejlesztése során kulcsfeladat volt a modularitás megvalósítása, mivel a feldolgozó osztályok a Labtech Kft. CardioSpy nevű szoftverébe kerültek integrálásra. A felvételek feldolgozását és a grafikus megjelenítést teljes mértékben függetlenítettük egymástól.

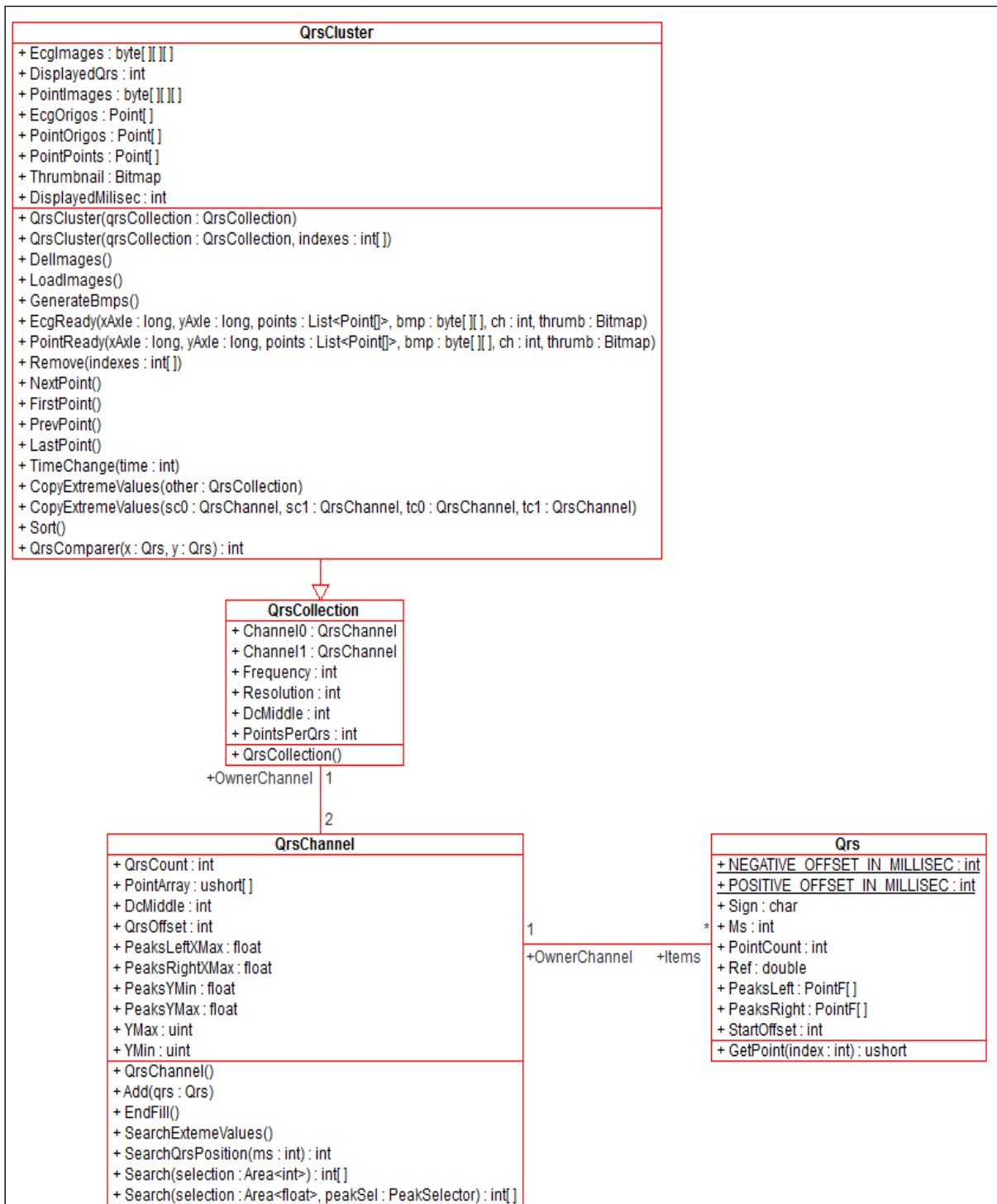
A felvételt feldolgozó osztályoknál elsődleges szempontok voltak a sebesség, a bővíthetőség, a jó paramétereizhetőség, az optimális erőforrás-gazdálkodás valamint a program többi részétől való függetlenség.

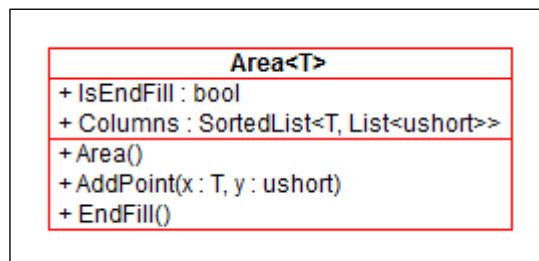
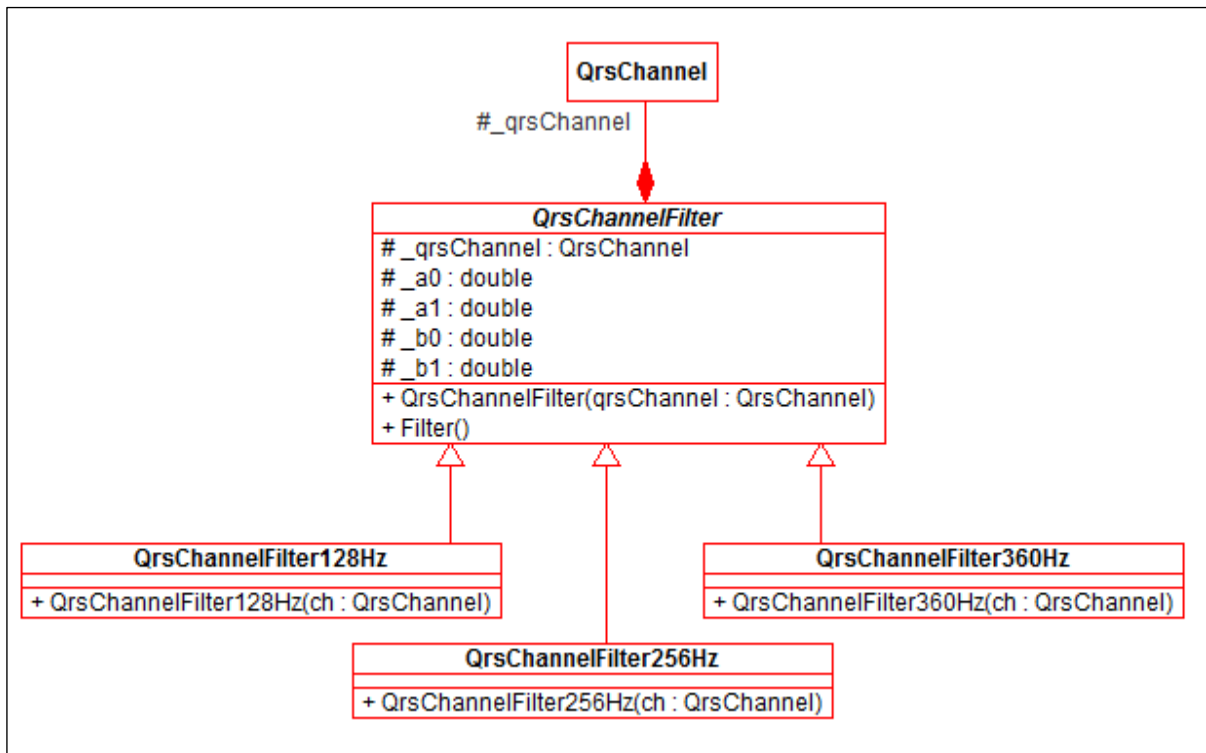
A megjelenítés feladatai között fontos volt a flexibilitás, a könnyű alakíthatóság, a nagyon nagy megjelenítendő képek (~2000x6000 pixel) valós idejű kezelése, a kicsinyítés/nagyítás képpont-veszteség nélküli megoldása, a megjelenített klaszterek szemléletessé tétele valamint a képek mozgatásának megvalósítása.

A program osztályszerkezete a következő UML diagramokon látható:

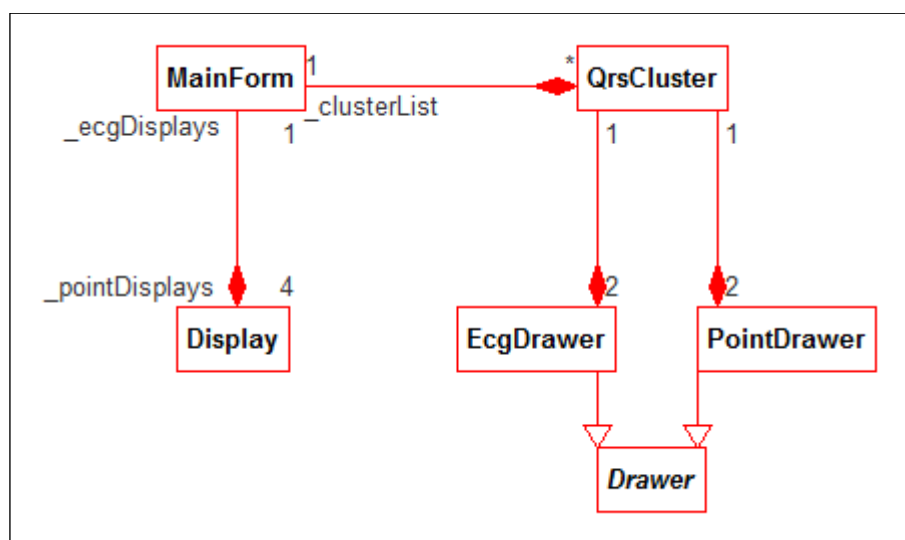
## 4.1 Adattároló és feldolgozó osztályok

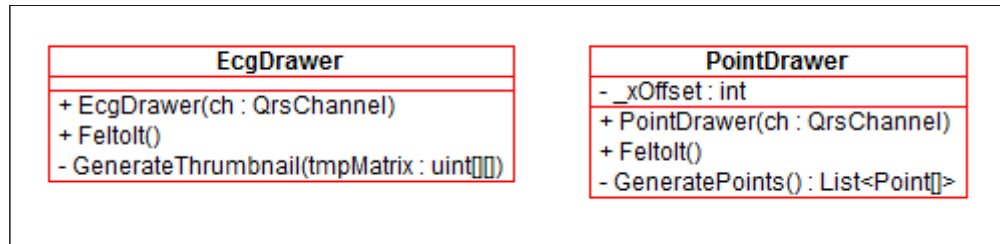






## 4.2 Megjelenítést végző osztályok





## Display

```
+ _zoomConversionValue : float = 800
- _canvas : System.Windows.Forms.PictureBox
- _vScroll : System.Windows.Forms.VScrollBar
- _hScroll : System.Windows.Forms.HScrollBar
- _xZoomBar : undef
- _yZoomBar : System.Windows.Forms.TrackBar
- _points : Point[]
- _imgMatrix : byte[][]
- _poly : List<Point>
- _xLineSteep : double
- _yLineSteep : double
- _origoX : double
- _origoY : double
- _xZoom : double
- _yZoom : double
- _hVal : double
- _vVal : double
- _paintedWidth : double
- _paintedHeight : double
- _xDifference : double
- _yDifference : double
- _canvasOrigoX : int
- _canvasOrigoY : int
+ AreaComplete : event AreaCompletedDelegate
+ HScrollModified : event ScrollModifiedDelegate
+ Display(displayImage : byte[], Origo : Point, hScroll : HScrollBar, xZoom : TrackBar, yZoom : TrackBar)
- DisplayResize(sender : object, e : EventArgs)
+ ChangeImage(displayImage : byte[], Origo : Point)
+ ChangePoints(points : Point[])
- Canvas_MouseDown(sender : object, e : MouseEventArgs)
- Canvas_MouseMove(sender : object, e : MouseEventArgs)
- Canvas_MouseUp(sender : object, e : MouseEventArgs)
- Lining(image : Bitmap, area : Rectangle, clear : bool)
- Crossing(image : Bitmap, area : Rectangle)
- GetColor(x : double, y : double) : byte
+ Draw()
- GlobalToCanvasX(global : double) : double
- GlobalToCanvasY(global : double) : double
- CanvasToGlobalX(canvas : double) : double
- CanvasToGlobalY(canvas : double) : double
+ XZoom(sender : object, e : EventArgs)
+ YZoom(sender : object, e : EventArgs)
- RedrawSection(image : Bitmap, area : Rectangle, clear : bool)
- MoveImage(xDifference : int, xNegative : bool, yDifference : int, yNegative : bool)
+ ScrollBarScrolled()
+ HScrollBarScrolled(sender : object, e : ScrollEventArgs)
- VScrollBarScrolled(sender : object, e : ScrollEventArgs)
```

## **4.3 Az osztályok rövid leírása**

### **4.3.1 Labtech.EcgViews névtér**

#### ***4.3.1.1 MainForm osztály (Balázs Ádám)***

Az osztály az alkalmazás főablakát reprezentálja, valamint fenntartja a kapcsolatot az objektumok között.

### **4.3.2 Labtech.EcgViews.Ecg névtér**

#### ***4.3.2.1 Area<T> osztály (Kuk József)***

Az osztály példánya egy kijelölést reprezentál. Az osztály képes kezelni bármilyen formájú (szabadkézi, konvex/konkáv poligon) kijelölést.

#### ***4.3.2.2 PeakSearcher osztály (Kuk József)***

Az osztály példánya referenciapontot és szélsőértékeket keres a csatornához tartozó szűrt jelen (FLT állományon).

#### ***4.3.2.3 Qrs osztály (Kuk József)***

Az osztály példánya egy QRS-t reprezentál.

#### ***4.3.2.4 QrsChannel osztály (Kuk József)***

Az osztály példánya a HOLTER felvétel egy csatornáját reprezentálja.

#### ***4.3.2.5 QrsCluster osztály (Balázs Ádám, Kuk József)***

Az osztály példánya egy a grafikus felhasználói felületen megjelenő klasztert reprezentál.

#### ***4.3.2.6 QrsCollection osztály (Kuk József)***

Az osztály példánya QRS-ek csatornába sorolt kollekcója.

#### ***4.3.2.7 QrsReader osztály (Kuk József)***

Az osztály példánya beolvassa a bináris, az annotációs és az FLT vagy ANT állományokat, melyek alapján feltölt egy QrsCollection példányt.

### **4.3.3 Labtech.EcgViews.Ecg.Filters névtér**

#### **4.3.3.1 QrsChannelFilter osztály (Kuk József)**

Absztrakt szűrő osztály.

#### **4.3.3.2 QrsChannelFilter128Hz osztály (Kuk József)**

Az osztály példánya egy 128Hz-es felüláteresztő szűrőt reprezentál.

#### **4.3.3.3 QrsChannelFilter256Hz osztály (Kuk József)**

Az osztály példánya egy 256Hz-es felüláteresztő szűrőt reprezentál.

#### **4.3.3.4 QrsChannelFilter360Hz osztály (Kuk József)**

Az osztály példánya egy 360Hz-es felüláteresztő szűrőt reprezentál.

### **4.3.4 Labtech.EcgViews.EcgDraw névtér**

#### **4.3.4.1 Drawer osztály (Balázs Ádám)**

Az absztrakt osztály a rajzoló osztályok közös tulajdonságait fogja egybe, valamint megteremti a kommunikációs felületet a többi osztállyal.

#### **4.3.4.2 EcgDrawer osztály (Balázs Ádám)**

Egy klaszter QRS-einek egymásra rajzolásával képet előállító osztály.

#### **4.3.4.3 PointDrawer osztály (Balázs Ádám)**

Egy csatorna szélsőérték pontjainak rajzolásáért felelős osztály.

### **4.3.5 Labtech.EcgViews.UI névtér**

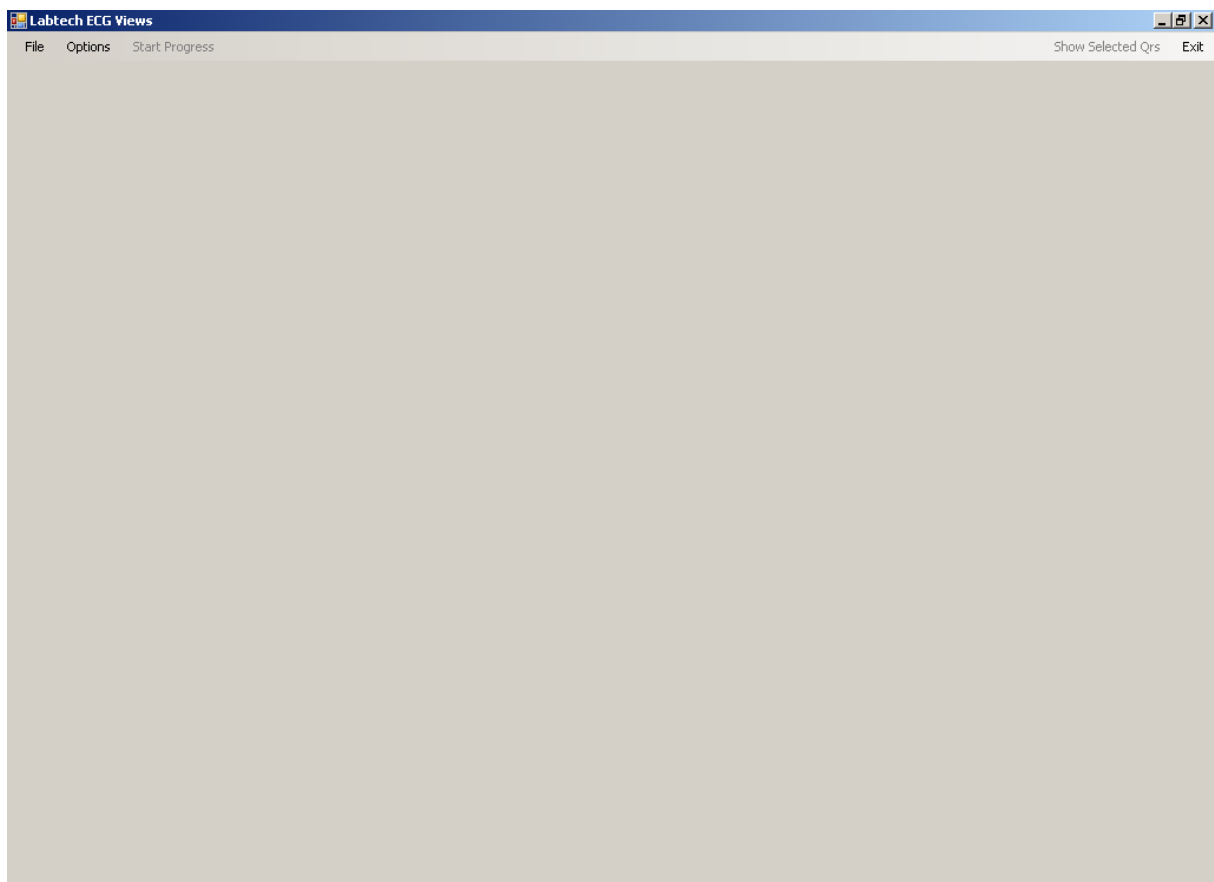
#### **4.3.5.1 Display osztály (Balázs Ádám)**

Egy indexelt bájtömbben reprezentált kép és koordináta-rendszerének megjelenítéséért felelős osztály.

## 5. Felhasználói Dokumentáció

### 5.1 Főablak

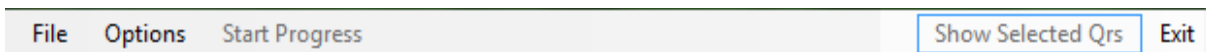
A program elindítása után annak főablaka jelenik meg (5.1. ábra).



5.1. ábra: Az alkalmazás főablaka

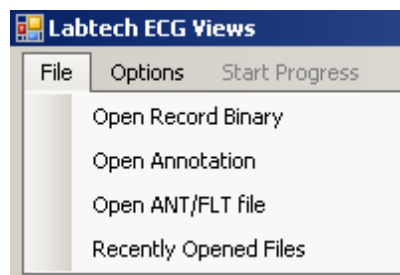
### 5.1.1 A menüsor elemei

Az alkalmazás menüsorának (5.2. ábra) elemei, azok funkciói:



5.2. ábra: Az alkalmazás menüSORA

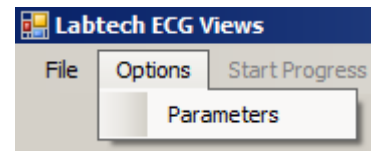
- **File menü** (5.3. ábra): A File menü alatt lenyíló menüpontok az alkalmazás által kezelt állományok kijelölésére alkalmasak.



5.3. ábra: A File menü elemei

- *Open Record Binary*: Egy Holter készülék által rögzített, szabványos fájlformátumú felvétel (.BIN állomány) kijelölése a program számára.
- *Open Annotation*: Egy szabványos formátumú .ANC állomány kijelölésére szolgáló menüpont, mely állomány az adott felvétel szívütéseinek helyét tartalmazza.
- *Open ANT/FLT file*: A menüpont alatt megjelenő fájlmegnyitási ablak segítségével a felhasználónak lehetősége van FLT és ANT állományok kijelölésére a programnak.
  - Az FLT állományok az adott felvétel egy szűrt reprezentációját tartalmazzák, melyet az alkalmazás feldolgoz, majd a belőle kinyert információk alapján szélsőérték pontpárokat alkot. Amennyiben FLT állomány kerül kijelölésre, a program ezen pontok segítségével valósítja meg a klaszterezést.
  - Az ANT állomány minden a felvételen megtalálható szívütéshez tartalmaz egy rá jellemző síkbeli pontot. Amennyiben a felhasználó ANT állományt jelöl ki, az alkalmazás ezen pontok segítségével valósítja meg a klaszterezési eljárást.
- *Recently Opened Files*: A menüpont az előzőleg a programban már megnyitott állományhármasok gyors elérési hivatkozásait tartalmazza, amennyiben van ilyen. Egy listabeli elem kiválasztása egyenértékű a felvétel (.BIN), és a hozzá tartozó segédállományok (.ANC és .FLT vagy .ANT) állományok kijelölésével.

- **Options menü** (5.4. ábra): Az Options menü alatt a program beállításait módosító menüpontok találhatók.



5.4. ábra: Az Options menü

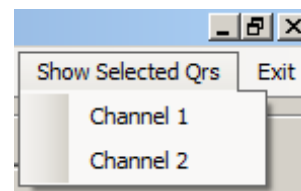
- **Parameters:** A program felvételek kezelésének, algoritmikus és klaszterezési paramétereinek beállításai módosítására szolgáló ablak. A beállítási lehetőségekről bővebben az 5.2. fejezet ír.

- **Start Progress menüpont:** A bemeneti állományok (.BIN, .ANC és .FLT vagy .ANT) specifikálása után aktívvá váló menüpont a kijelölt fájlok feldolgozását indítja el. A feldolgozást egy folyamatjelző mutatja (5.5. ábra). A művelet befejezését követően láthatóvá válnak a felvétel nézeteit mutató megjelenítők, melyek leírásai az 5.3.2.4 valamint az 5.3.2.5 fejezetben találhatóak.



5.5. ábra: A felvétel feldolgozását mutató folyamatjelző

- **Show Selected Qrs menü** (5.6. ábra): Miután egy felvétel feldolgozásra került, a menü aktívvá válik. Segítségével lehetőség van az éppen kijelölt QRS (5.3.2.2 fejezet) megjelenítésére.

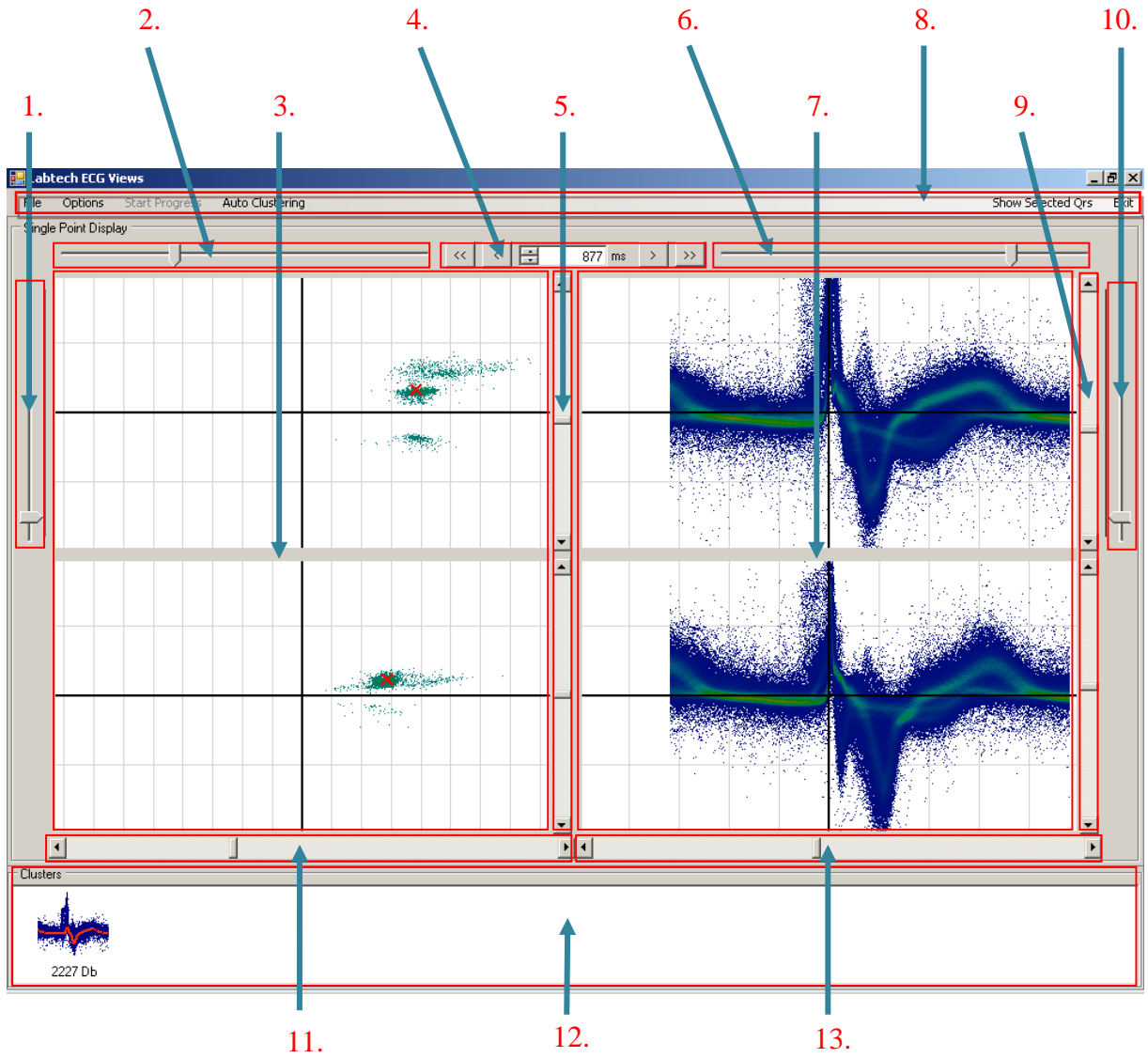


5.6. ábra: A Show Selected Qrs menü elemei

- **Exit menüpont:** A menüpontot aktiválva az alkalmazás szabályosan befejezi a futását.

### 5.1.2 A program főablaka a fájlfeldolgozás után

A fájlfeldolgozás befejeztével az alkalmazás főablaka az 5.7-es ábra szerint változik.



11. 12. 13.  
5.7. ábra: A program főablaka a fájlfeldolgozás után

### **5.1.3 Az 5.7. ábra jelmagyarázata:**

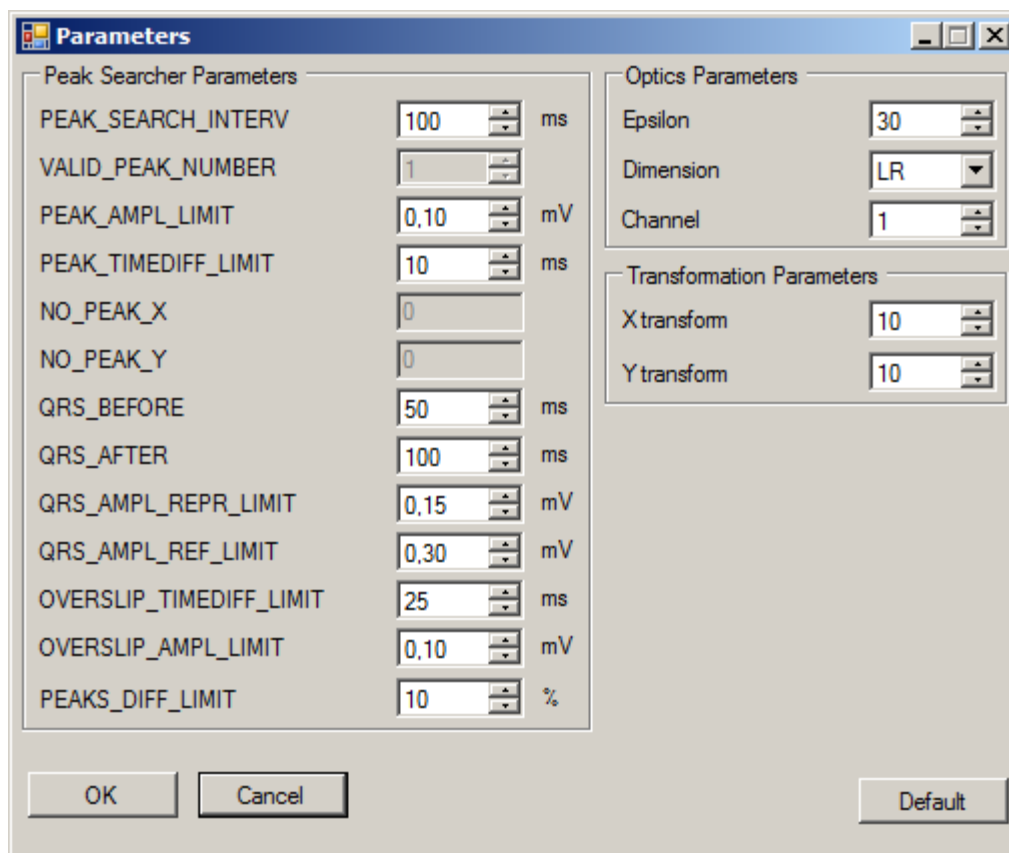
1. A pontfelhő megjelenítők függőleges nagyítását lehetővé tevő csúszka. A két csatorna nagyításáért ugyanaz a csúszka felelős.
2. A pontfelhő megjelenítők vízszintes nagyítását lehetővé tevő csúszka. A két csatorna nagyításáért ugyanaz a csúszka felelős.
3. A pontfelhő megjelenítők.
4. Az aktuális klaszteren belül kijelölt QRS pozícióját léptető mező.
5. A pontfelhő megjelenítők képeinek függőleges pozícióit beállító csúszkák. A két csatorna pozícióit a két csúszka külön-külön kezeli.
6. A QRS megjelenítők vízszintes nagyítását lehetővé tevő csúszka. A két csatorna nagyításáért ugyanaz a csúszka felelős.
7. A QRS megjelenítők.
8. A főablak menüsora.
9. A QRS megjelenítők képeinek függőleges pozícióit beállító csúszkák. A két csatorna pozícióit a két csúszka külön-külön kezeli.
10. A QRS megjelenítők függőleges nagyítását végző csúszka. A két csatorna nagyításáért ugyanaz a csúszka felelős.
11. A pontfelhő megjelenítők képeinek vízszintes pozícióit beállító csúszka. A két csatorna pozíciójáért ugyanaz a csúszka felelős.
12. A klasztereket és azok template-eit ábrázoló miniatűr képek listája, melyek segítségével váltani lehet az aktuálisan ábrázolt klaszterek között.
13. A QRS megjelenítők képeinek vízszintes pozícióját beállító csúszkák. A két csatorna pozíciójáért ugyanaz a csúszka felelős.

## 5.2 A program paramétereinek beállítása

A program paramétereinek módosítása az *Options* menü *Parameters* menüpontjának kiválasztásával lehetséges (5.8. ábra).

A megjelenő panelen elvégezhető beállítások függenek attól, hogy a programot aktuálisan FLT vagy ANT inputtal használjuk.

FLT állomány használata esetén, az abban található *szűrt jelen* a program által végzett speciális szélsőérték-keresés eredményeképp minden egyes QRS-hez egy kétdimenziós pontpár határozódik meg. ANT állomány használatakor a benne található (egy külső szoftver segítségével előállított) numerikus információk feldolgozásával kerül meghatározásra a QRS-ekhez egy darab síkbeli pont. Az így meghatározott pontpár/pont jelenik meg a program főablakának *pontfelhő megjelenítőjén*, illetve képzik a klaszterezés alapját. Amennyiben FLT állományt használunk, a pontpárok meghatározása részletesen paraméterezhető, ezért ekkor több beállítási lehetőség áll rendelkezésre.



5.8. ábra: A program paramétereit módosító ablak

A kívánt beállítások elvégzése után az *OK* gombra való kattintással a beállítások elfogadhatók, a *Cancel* gomb kiválasztásával elvethetők, illetve a *Default* gomb megnyomásával az alapértelmezett paraméterek állítódnak vissza.

## 5.2.1 Paraméterek megadása FLT állomány használata esetén

### 5.2.1.1 Szélsőérték-keresés paramétere (5.9. ábra)

A szélsőérték-keresés paramétereinek meghatározására csak FLT állomány használata esetén van szükség, lehetőség. A paraméterek azt befolyásolják, hogy az egyes QRS-ekhez a szűrt jelen (FLT állomány) hogyan határozódnak meg a szélsőérték-keresés alapjául szolgáló pontpárok.

Az egyes paraméterek jelentése:

#### *PEAK\_SEARCH\_INTERV:*

Meghatározza, hogy a szűrt jelen mekkora tartományban történik a szélsőértékek keresése. Nagyobb érték esetén szélesebb tartományban megy végbe a keresés.

#### *VALID\_PEAK\_NUMBER:*

Meghatározza, hogy egy-egy oldalról hány szélsőérték pontot keresünk. A paraméter alapértelmezett értéke 1, nem módosítható.

#### *PEAK\_AMPL\_LIMIT:*

Zaj magasság határ. A szűrt jelen előfordulhatnak kisebb zajok, melyek a keresést negatívan befolyásolnák. Ez a paraméter meghatározza, hogy mekkora az az amplitúdóbeli kilengés, amit zajnak tekintünk. A megadott milli volt érték alatti eltéréseket figyelmen kívül hagyjuk a szélsőértékek meghatározása szempontjából.

Parameter	Value	Unit
PEAK_SEARCH_INTERV	100	ms
VALID_PEAK_NUMBER	1	
PEAK_AMPL_LIMIT	0.10	mV
PEAK_TIMEDIFF_LIMIT	10	ms
NO_PEAK_X	0	
NO_PEAK_Y	0	
QRS_BEFORE	50	ms
QRS_AFTER	100	ms
QRS_AMPL_REPR_LIMIT	0.15	mV
QRS_AMPL_REF_LIMIT	0.30	mV
OVERSLIP_TIMEDIFF_LIMIT	25	ms
OVERSLIP_AMPL_LIMIT	0.10	mV
PEAKS_DIFF_LIMIT	10	%

5.9. ábra: A szélsőérték-keresés paramétere

*PEAK\_TIMEDIFF\_LIMIT:*

A paraméter a PEAK\_AMPL\_LIMIT paraméter vízszintes irányú (időbeli) megfelelője.

*NO\_PEAK\_X, NO\_PEAK\_Y:*

Nem minden esetben határozható meg szélsőérték. Ilyen esetekben ez az érték kerül beírásra.

A paraméter a felhasználó által nem módosítható.

*QRS\_BEFORE, QRS\_AFTER, QRS\_AMPL\_REPR\_LIMIT, QRS\_AMPL\_REF\_LIMIT:*

Technikai jellegű paraméterek, a felhasználó számára az alapértelmezett értékek módosítása nem ajánlott. Amennyiben a szűrt jelen a QRS jelzés előtt QRS\_BEFORE által, vagy után QRS\_AFTER által meghatározott tartományban az amplitúdó különbség kisebb, mint

- a) QRS\_AMPL\_REF\_LIMIT, akkor a csatornán nem lesz referenciapont keresés és a másik csatorna referenciapontját használjuk, vagy
- b) QRS\_AMPL\_REPR\_LIMIT, referenciapont és szélsőérték-keresés sem kis amplitúdó miatt.

*OVERSLIP\_TIMEDIFF\_LIMIT, OVERSLIP\_AMPL\_LIMIT:*

A paraméterek segítségével megakadályozhatóak az úgynevezett túlcúsúzó szélsőértékek megjelenése. Amennyiben a szűrt jelen egy szélsőérték meghatározása folyamán egy pont utáni OVERSLIP\_TIMEDIFF\_LIMIT által meghatározott tartományon belül nincs az egyes pontok között legalább OVERSLIP\_AMPL\_LIMIT által meghatározott amplitúdó különbség, az adott pontot tekintjük szélsőértéknek.

*PEAKS\_DIFF\_LIMIT:*

Az úgynevezett dupla szárú szűrt jellel rendelkező QRS-ek esetében a szélsőérték meghatározását speciális módon kell elvégezni. Ez a paraméter meghatározza, hogy az egyes szárok magassága között legalább hány százalékos eltérésnek kell lennie. Ha az eltérés ennél kisebb, a szélsőérték-keresést speciálisan végezzük.

### ***5.2.1.2 Automatikus klaszterezés paramétere***

FLT állomány használata esetén automatikus klaszterezésre nincs lehetőség.

## 5.2.2 Paraméterek megadása ANT állomány használata esetén

### 5.2.2.1 Szélsőérték-keresés paramétereit

ANT állomány használatakor a pontok meghatározása a szoftver hatáskörén kívül esik, annak paramétereit közvetlenül nem módosíthatóak.

### 5.2.2.2 Automatikus klaszterezés paramétereit (5.10. ábra)

A különböző formájú QRS-ek szétválasztása ANT állomány használatakor, az egyes szívtütesekhez – az ANT állományból kiolvasott – kétdimenziós pontok alapján történik. Az egy csoportban lévő ponthalmazokhoz tartozó QRS-ek formája orvosi szempontból hasonlóan tekinthető.

Optics Parameters	
Epsilon	30
Dimension	LR
Channel	1

Transformation Parameters	
X transform	10
Y transform	10

5.10. ábra: Az automatikus klaszterezés paramétereit

Az automatikus klaszterezés paramétereitvel befolyásolható, hogy a pontok csoportokra bontása milyen részletességgel történjen.

*X transform, Y transform:*

A klaszterezés sebességének növelése érdekében a klaszterezés megkezdése előtt egy előzetes hasonlósági transzformációt hajtunk végre. A transzformáció során a pontokra egy rácsot (gridet) illesztünk, melynek metszéspontjaiba húzzuk a környezetében lévő pontokat.

A két paraméter ennek a gridnek a X (vízszintes) és Y (függőleges) irányú felbontását szabályozzák. Nagyobb érték megadása esetén a rácsozat felbontása durvább. Nullás érték megadása esetén nem történik transzformáció.

*Epsilon:*

A paraméter az egyes ponthalmazok csoportokra bontásának finomságát szabályozza. Nagyobb érték esetén a finomság csökken.

*Dimension:*

Használton kívüli paraméter.

*Channel:*

Meghatározza, hogy melyik EKG csatorna képi a QRS-ek klaszterezésének alapját.

## 5.3 A program használata

### 5.3.1 A program használatának megkezdése, a felvétel kiválasztása

A program elindítása után annak főablaka jelenik meg (5.1. ábra). A főablak menüsorában, a File menüben ki kell választanunk a feldolgozni kívánt felvételt. Erre két lehetőség is van:

- Az *Open Record Binary*, *Open Annotation* és *Open ANT/FLT file* menüpontok segítségével külön-külön megadjuk a felvételhez tartozó állományokat (*részletesen 5.1.1 fejezet*)
- A *Recently Opened Files* alatt található listából kiválasztható a programmal előzőleg már megnyitott állományok közül az, amelyet fel szeretnénk dolgozni. Ezzel az összes, a felvételhez tartozó állomány kiválasztásra kerül. Ez a lehetőség csak akkor elérhető, ha az éppen feldolgozni kívánt felvétel már korábban meg volt nyitva a programmal.

A felvétel specifikálása után a főmenü Start Progress menüpontja elérhetővé válik. A Start Progress menüpontot kiválasztva az alkalmazás megkezdi a felvétel feldolgozását. A feldolgozás menetét egy, a Start Progress gomb helyén megjelenő folyamatjelző mutatja.

### 5.3.2 A főablakon végezhető műveletek

Miután a program feldolgozta a kijelölt állományokat, a feldolgozás utáni főablak lesz látható (5.7 ábra).

#### 5.3.2.1 Klaszterek közötti váltás

A főablakon lévő Pontfelhő- és QRS megjelenítők a feldolgozott felvétel egy, felhasználó által kijelölt *klaszterét* jelenítik meg. *Klaszternek* nevezzük a QRS-ek jellemző tulajdonságaik alapján összetartozónak ítélt csoportjait. A feldolgozás befejezését követően a felvétel összes QRS-e egy kezdeti, inicializációs klaszterbe kerül. A felvételen létrehozott klaszterek a program ablakának alján, a „Clusters” szekcióban jelennek meg, a megjelenítőkön aktuálisan ábrázolt klaszter körül kijelölés található. „Clusters” alatt megjelenített bélyegképek a klaszter QRS-einek egymásra rajzolt képének kicsinyített változatát ábrázolják, a narancssárga kiemelt görbe pedig klaszterhez tartozó template. A klaszterek közötti váltás a bélyegképekre való kattintással történik.

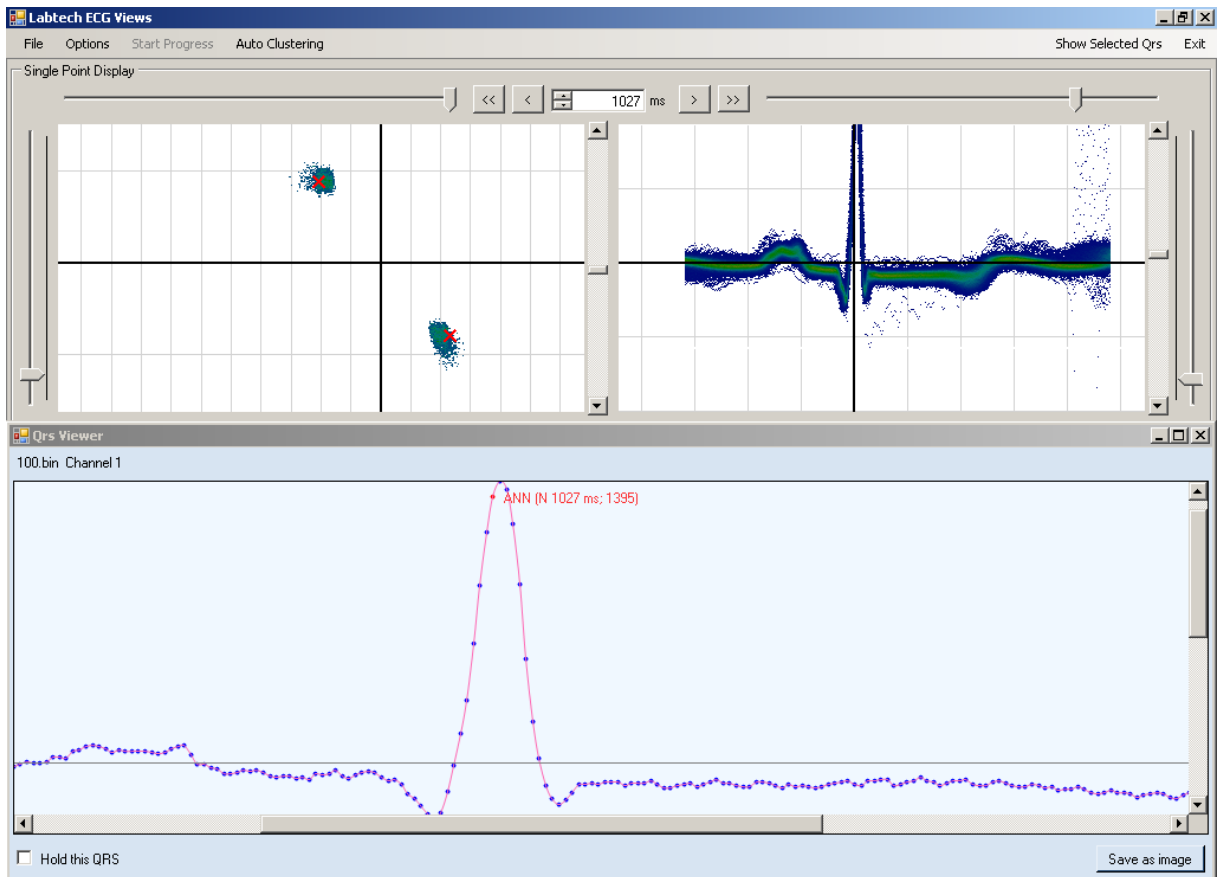
### **5.3.2.2 Klaszteren belüli QRS kiválasztás**

A megjelenítőkön ábrázolt klaszterbeli QRS-ek között lehetőség van egy aktuális szívütést kijelölni. Ezt a főablak tetején, középen lévő pozicionáló mező segítségével lehet megtenni. A bal szélső, valamint a jobb szélső („<<” és „>>” feliratú) gomb megnyomásával az aktuális QRS pozícióját a klaszter időben legelső, valamint legutolsó QRS-ére lehet állítani. A két belső („<” és „>” feliratú) gomb segítségével az éppen kijelölt QRS-hez az aktuális klaszteren belül időben egyel korábbi, valamint egyel későbbi QRS-re lehet váltani. A középen található mező az éppen kijelölt QRS a teljes felvétel elejétől vett időbeni távolságát mutatja milliszekundumban. Lehetőség van a mezőbe való direkt adatbevitelre is, ha az adott klaszteren belüli QRS-ek közül egy adott időponthoz legközelebbi szívütést keresünk, a mezőbe be kell írunk az adott időpontot milliszekundumban, majd entert nyomva a program a kijelölt QRS-t erre állítja. Ha a megadott időpontban nincs QRS, akkor az adott időpillanathoz legközelebbi, kijelölt klaszteren belüli QRS kerül kiválasztásra.

### **5.3.2.3 A kijelölt QRS elemzésének lehetőségei**

A klaszterben kijelölt QRS szélsőérték pontja kiemelésre kerül a pontfelhő megjelenítőkön (5.3.2.5. fejezet).

A főmenü „**Show Selected Qrs**” menüjében lehetőség van a kijelölt QRS egyes vagy kettes csatornabeli képének megjelenítésére.



5.11. ábra: Kijelölt QRS megjelenítése

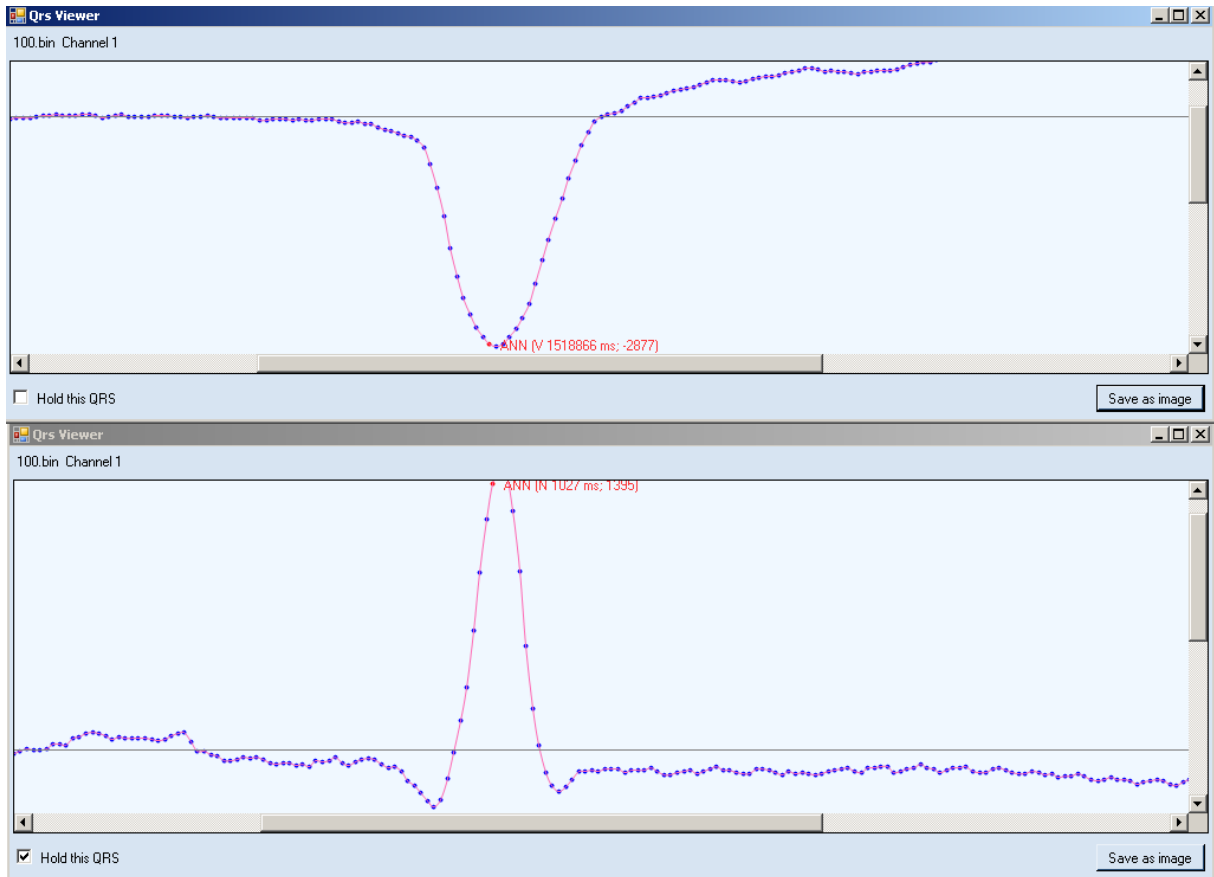
A megjelenített QRS egy új ablakban kerül megjelenítésre (5.11. ábra). A megnyíló ablak felső paneljén a QRS eredeti képe látható. Amennyiben FLT felvételt használunk, az ablak alsó részén a kiválasztott QRS szűrt jele látható.

Az ablak megnyitása után lehetőség van tovább dolgozni az alkalmazás főablakában. Amennyiben a főablakban a kiválasztott QRS-t tovább léptetjük, a megnyitott ablakban is az új QRS kerül megjelenítésre.

Kivétel ez alól, hogyha a megnyitott QRS nézegető ablak alján található „Hold this QRS” jelölőnégyzet aktiválva van. Ebben az esetben az ablakban megjelenített QRS mindig a jelölőnégyzet aktiválásakor megjelenített szívütés marad.

Lehetőség van több QRS megjelenítő ablak megnyitására is. Ez a funkció felhasználható a különböző QRS-ek összehasonlítására. Ehhez ki kell választani az első összehasonlítani kívánt szívütést a főablakban. A „**Show Selected QRS**” menü segítségével megjelenített ablak alján a „Hold this QRS” jelölőnégyzet segítségével rögzítjük az ablakon megjelenített

szívütést. A főablakba visszatérve kiválasztható a következő, összehasonlítani kívánt QRS. A kiválasztott QRS-t ezután megjeleníthetjük a „**Show Selected QRS**” menü segítségével, melynek hatására egy új QRS megjelenítő ablak jelenik meg. Így könnyen összehasonlíthatóak a szívütések (5.12. ábra).



5.12. ábra: Két QRS összehasonlítása

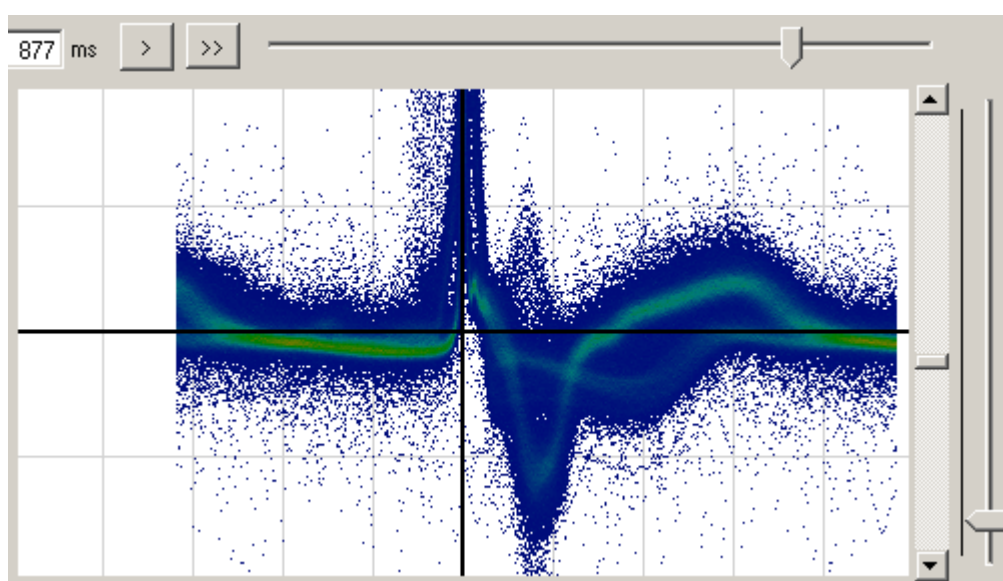
A QRS megjelenítő ablak alján található „Save as image” gomb segítségével a megjelenítőn látható ábra kép formátumban is elmenthető.

#### 5.3.2.4 A QRS megjelenítők

A QRS megjelenítőkön a kiválasztott klaszter szívütései vannak ábrázolva. A megjelenítő koordináta rendszerének X tengelye az idő, Y tengelye pedig a Holter készülék által mért elektromos feszültség. A klaszterben lévő QRS-ek *annotációs pontjai* rögzítésre kerülnek az idő tengely nulla pontjában. Ehhez képest kerül meghatározásra az ábrázolt tartomány, ami az annotáció előtti 250 milliszekundum, valamint az annotáció utáni 300 milliszekundum. A klaszterben lévő minden egyes QRS ezen időablakba eső része kerül ábrázolásra a QRS megjelenítőn. Az idő tengely negatív és pozitív tartományában az adott QRS annotáció előtti

és utáni időpillanatokban felvett feszültség értékei kerülnek ábrázolásra. A különböző, kiválasztott klaszterbeli QRS-ek képei egymásra rajzolódva jelennek meg.

Ha több QRS egy bármely X tengelybeli pontjához ugyanazon feszültség-érték tartozik, akkor a koordináta-rendszer e pontjának a sűrűsége annyi lesz, ahány QRS áthalad az adott ponton. A megjelenítés során ezek a sűrűség-értékek színértékek segítségével kerülnek ábrázolásra. A kék színű pontok a többi, a megjelenítőn szereplő ponthoz képest a legalacsonyabb sűrűségűek, míg a piros színű értékek a többi ponthoz képest a legmagasabb sűrűségűeket, a színskála e kettő közötti értékei pedig a köztes sűrűségűeket jelölik.



5.13. ábra: A felvétel egyik csatornája a QRS megjelenítőn

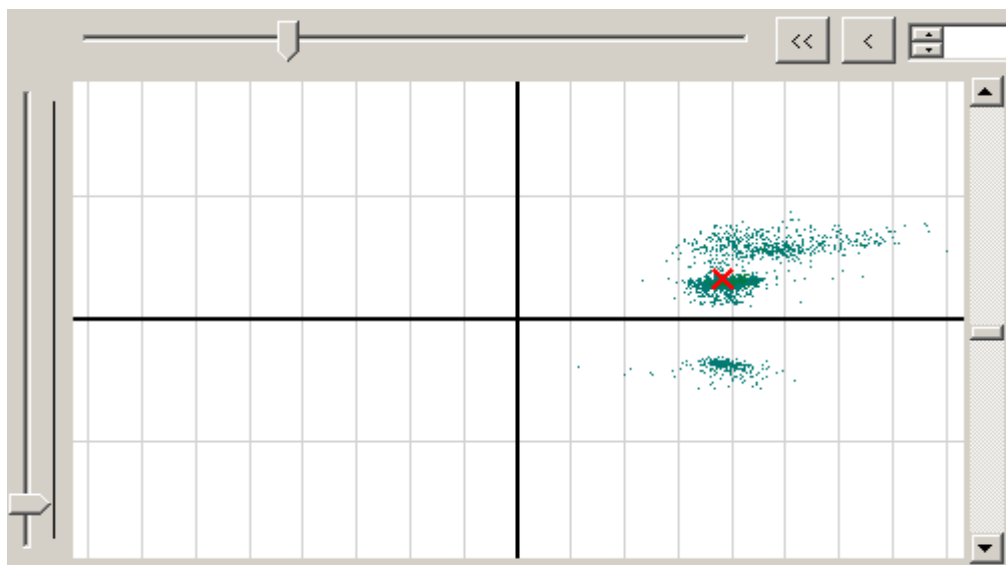
A főablakon két QRS megjelenítő található, a felső a felvétel egyes, az alsó a kettes csatornáját ábrázolja. Egy csatorna az 5.13. ábrán látható módon kerül megjelenítésre.

#### 5.3.2.5 A Pontfelhő megjelenítők

A pontfelhő megjelenítőkön FLT állomány feldolgozása esetén a kiválasztott klaszter szívütéseinek a szívütés *annotációitól* számított speciális szélsőérték-pontpárjai vannak ábrázolva. ANT fájl feldolgozása esetén az egy-egy szívütésre jellemző *Wavelet pont* kerül ábrázolásra az összes, kiválasztott klaszterbeli QRS-re. A megjelenítő koordináta rendszerének X tengelye az idő, Y tengelye pedig a Holter készülék által mért elektromos feszültség. A klaszterben lévő QRS-ek szélsőérték- vagy wavelet pontjai az egyes QRS-ekre jellemző annotációs ponttól való távolságuk szerint kerülnek ábrázolásra. Az annotációs

pontok az idő tengely nulla pontjába kerülnek, ehhez képest kerülnek ábrázolásra a szélsőérték- vagy wavelet pontok. Az idő tengely negatív és pozitív tartományában az adott QRS annotáció előtti és utáni időpontokba eső szélsőértékei vagy wavelet pontjai kerülnek ábrázolásra. A különböző, kiválasztott klaszterbeli QRS-ek pontjai egymásra rajzolódva jelennek meg.

Ha több QRS pont egy bármely X tengelybeli pontjához ugyanazt a feszültség-értéket veszi fel, akkor a koordináta-rendszer e pontjának a sűrűsége annyi lesz, ahány QRS pont esik az adott pontba. A megjelenítés során ezek a sűrűség-értékek színértékek segítségével kerülnek ábrázolásra. A kék színű pontok a többi, a megjelenítőn szereplő ponthoz képest a legalacsonyabb sűrűségűek, míg a piros színű értékek a többi ponthoz képest a legmagasabb sűrűségértékeket, a színskála e kettő közötti értékei pedig a köztes sűrűségértékeket jelölik.



5.14. ábra: A felvétel egyik csatornája a QRS megjelenítőn

A pontfelhő-megjelenítőn a kiválasztott QRS szélsőérték- vagy wavelet pontjai a pontba eső piros 'X'-el kiemelésre kerülnek.

A főablakon két pontfelhő megjelenítő található, a felső a felvétel egyes, az alsó a kettes csatornáját ábrázolja. Egy csatorna pontfelhő megjelenítőn ábrázolt képe az 5.14. ábrán látható.

### **5.3.2.6 A megjelenítők működése**

A megjelenítők képei az X és az Y tengely mentén is nyújthatóak. A nagyításért felelős csúszkák egyaránt hatással vannak mindkét, azonos típusú (pontfelhő- és QRS) megjelenítőkre, így a két csatorna képe mindig azonos nagyítással kerül megjelenítésre. Az X tengely nyújtásáért a vízszintes, míg az Y tengelyéért a függőleges csúszka a felelős.

A megjelenítőkön lehetőség van a koordináta-rendszer eltolására is. Az azonos típusú megjelenítők Y tengelyeinek vízszintes elmozdításáért ugyanaz a csúszka felelős. Ennek eredményeképp a két megjelenítő Y tengelye mindig egy vonalban marad, ezzel megkönnyítve a feldolgozást. Ezzel a vízszintes csúszkával lehet az azonos típusú megjelenítők Y tengelyét mozgatni.

Az X tengelyek mozgatására azonban külön-külön van lehetőség. Minden megjelenítő mellett található egy függőleges csúszka, ennek segítségével lehet az X tengely pozícióját módosítani.

A megjelenítők koordináta-rendszere a csúszkákon kívül módosítható 'Fogd és Vidd' módszerrel is: Az egeret a megjelenítő fölé pozicionálva, majd a bal egérgombot nyomva tartva a koordináta-rendszer helyzete követni fogja az egér elmozdulását.

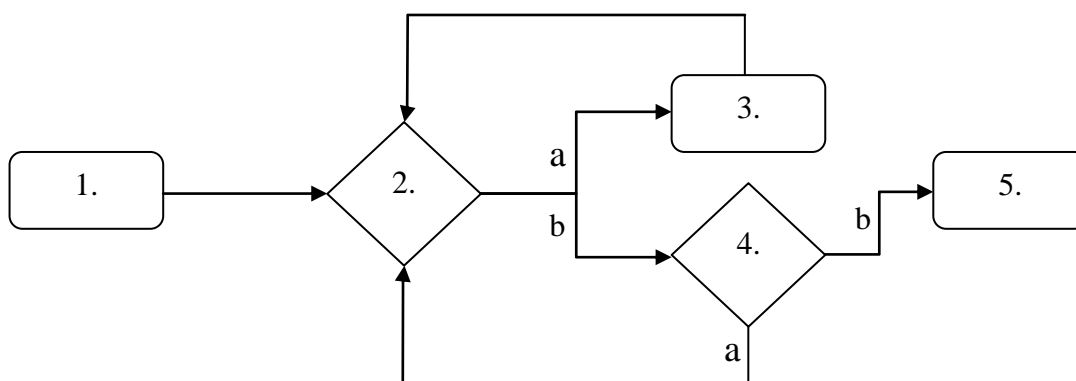
### **5.3.3 A kézi klaszterezés folyamata**

A kézi klaszterezés során a program a felhasználó instrukciói alapján rendeli klaszterekbe a QRS-eket. Kézi klaszterezéskor a felhasználónak kell megjelölni az aktuálisan kiválasztott klaszter valamely megjelenítőjén, hogy mely QRS-ek vagy szélsőérték/wavelet ponthoz tartozó szívütések kerüljenek egy új klaszterbe. A megadás minden esetben a megjelenítőkön egy terület kijelölésével történik. A kijelölés egy téglalap alapú terület. Az egeret a klaszterezéshez használni kívánt megjelenítő fölé pozicionálva a jobb egérgomb lenyomásával a kijelölő téglalap egyik csúcspontja rögzítésre kerül a lenyomás pillanatában lévő egérpozícióban. A jobb egérgombot folyamatosan nyomva tartva a kijelölő téglalap másik csúcspontja tovább mozgatható. Amennyiben a kijelölést az általunk kiválasztott terület fölé állítottuk, a jobb egérgomb felengedésével rögzítjük a kijelölés második csúcspontját, majd a program elvégzi a kijelölt QRS-ek új klaszterbe sorolását.

### 5.3.3.1 Klaszterezési technikák a QRS megjelenítőkön

A QRS megjelenítőkön történő klaszterezés a kiválasztott felvétel feldolgozásának egyik lehetősége. A feldolgozás az eredeti, majd a feldolgozás során keletkezett klaszterek egymás utáni átvizsgálásával történik. Ez az elemzési módszer kombinálható az automatikus klaszterezéssel (5.3.4 fejezet), valamint a pontfelhő megjelenítőkön történő klaszterezéssel is (5.3.3.2 fejezet). Lépései: (5.15. ábra)

1. Válasszuk ki a legelső klasztert.
2. Vizsgáljuk meg a kiválasztott klaszter QRS megjelenítőkön ábrázolt képét.
  - a. Amennyiben a QRS megjelenítők valamely csatornájának képen elkülöníthető, különböző formájú QRS-ek csoportjai fedezhetőek fel, ugorjunk a 3. pontra.
  - b. Amennyiben a QRS megjelenítők valamennyi csatornájának a képein csak hasonló formájú QRS-ek találhatók, ugorjunk a 4. pontra.
3. Az egyik elkülöníthető csoport QRS-eit helyezzük új klaszterbe. Ennek pontos menete a következő bekezdésben, valamint az 5.3.3 fejezetben kerül kifejtésre. A QRS-ek új klaszterbe helyezése után válasszuk ki azt a klasztert, amelyből az előbb eltávolítottuk a QRS-ek egy csoportját, majd ugorjunk a 2. pontra.
4. Vizsgáljuk meg, hogy van-e még nem elemzett klaszter.
  - a. Amennyiben van, válasszuk ki a következő klasztert, majd folytassuk a folyamatot a 2. pontban.
  - b. Amennyiben nincs több vizsgálandó klaszter, ugorjunk az 5. pontra.
5. A felvétel feldolgozása befejeződött.

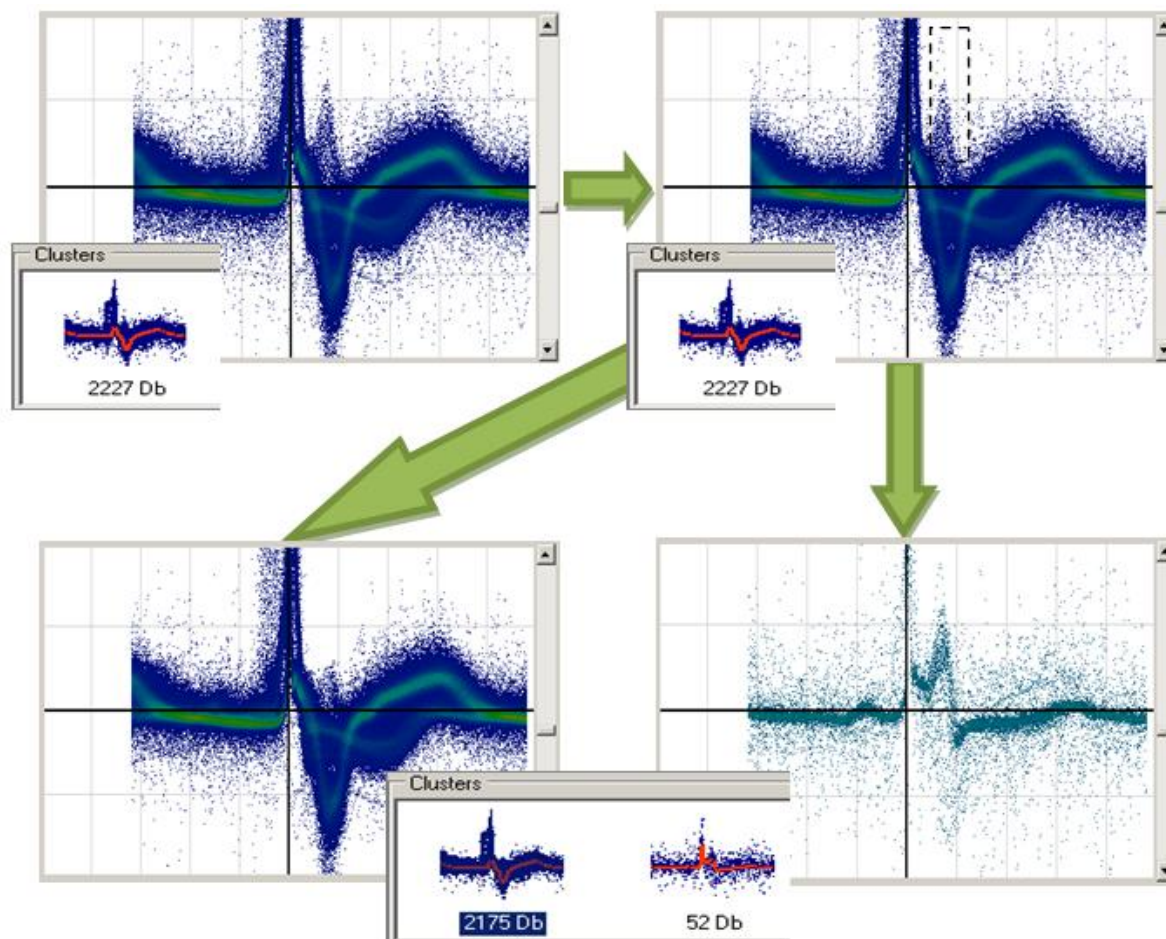


5.15. ábra: A Kézi klaszterezés folyamata a QRS megjelenítőn

A QRS megjelenítőn történő kézi klaszterezés során az 5.3.3 fejezetben leírt, az új klaszterbe átmozgatandó QRS-eket megadó kijelölést úgy kell megadni, hogy az összes QRS, amelynek

legalább egy pontja bele esik a kijelölésbe, az átkerül az új klaszterbe. Ennek megfelelően a kijelölést úgy kell megadni, hogy abba csak azon QRS-ek pontjai essenek, amelyeket az új klaszterbe szeretnénk mozgatni.

A kézi klaszterezés folyamata a QRS megjelenítőn a gyakorlatban az 5.16. ábrán látható.



5.16. ábra: A kézi klaszterezés megvalósítása a QRS megjelenítőn

### 5.3.3.2 Klaszterezési technikák a pontfelhő-megjelenítőkön

A pontfelhő megjelenítőkön történő klaszterezés a kiválasztott felvétel feldolgozásának másik lehetősége. A feldolgozás az eredeti, majd a feldolgozás során keletkezett klaszterek egymás utáni átvizsgálásával történik. Ez az elemzési módszer kombinálható az automatikus klaszterezéssel (5.3.4 fejezet), valamint a QRS megjelenítőkön történő klaszterezéssel is (5.3.3.1 fejezet). Lépései:

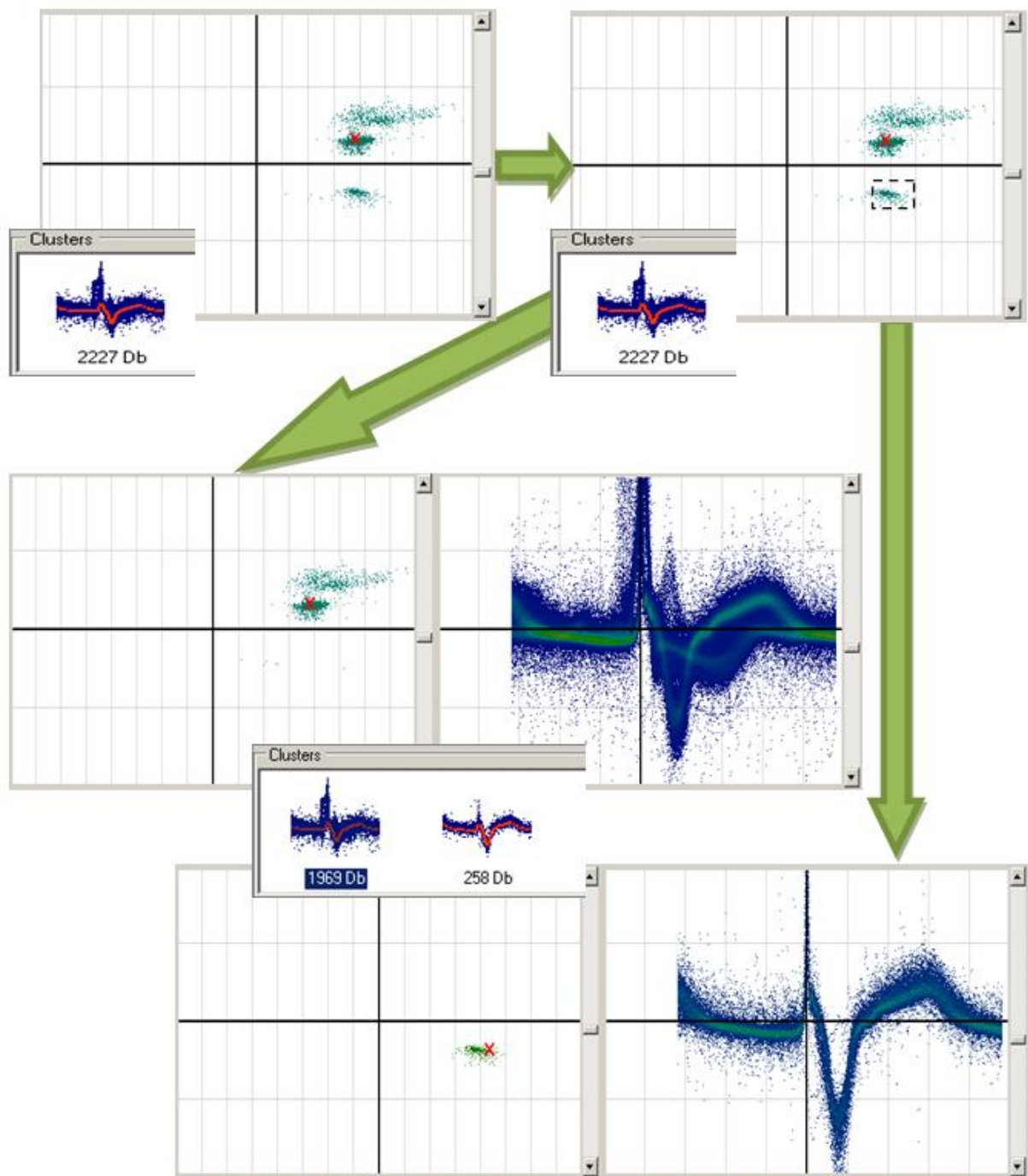
1. Válasszuk ki a legelső klasztert.

2. Vizsgáljuk meg a kiválasztott klaszter pontfelhő megjelenítőkön ábrázolt képét.
  - a. Amennyiben a pontfelhő megjelenítők valamely csatornájának képén jól elkülönülő pont-csoportok fedezhetőek fel, ugorjunk a 3. pontra.
  - b. Amennyiben a pontfelhő megjelenítők valamennyi csatornájának a képein csak egybefüggő, jól el nem határolható pontcsoportok találhatóak, ugorjunk a 4. pontra.
3. Az egyik elkülöníthető pontcsoport pontjait helyezzük új klaszterbe. Ennek pontos menete a következő bekezdésben, valamint az 5.3.3 fejezetben kerül kifejtésre. Az alkalmazás elvégzi a pontokhoz tartozó QRS-ek új klaszterbe helyezését, majd ezután válasszuk ki azt a klasztert, amelyből az előbb eltávolítottuk a pontok csoportját és ugorjunk a 2. pontra.
4. Vizsgáljuk meg, hogy van-e még nem elemzett klaszter?
  - a. Amennyiben van, válasszuk ki a következő klasztert, majd folytassuk a folyamatot a 2. pontban.
  - b. Amennyiben nincs több vizsgálandó klaszter, ugorjunk az 5. pontra.
5. A felvétel feldolgozása befejeződött.

A feldolgozás folyamatábrája megegyezik a 5.15 ábrán látható QRS megjelenítőkön végzett kézi klaszterezésével.

A pontfelhő megjelenítőn történő kézi klaszterezés során az 5.3.3 fejezetben leírt, az új klaszterbe átmozgatandó szélsőérték- vagy wavelet pontokhoz tartozó kijelölést úgy kell megadni, hogy abba az összes új klaszterbe átvinni kívánt pont beleessen. Ennek megfelelően a kijelölést úgy kell megadni, hogy abba csak azon pontok essenek, amelyekhez tartozó QRS-eket az új klaszterbe szeretnénk mozgatni.

A kézi klaszterezés folyamata a pontfelhő megjelenítőn a gyakorlatban az 5.17. ábrán látható.

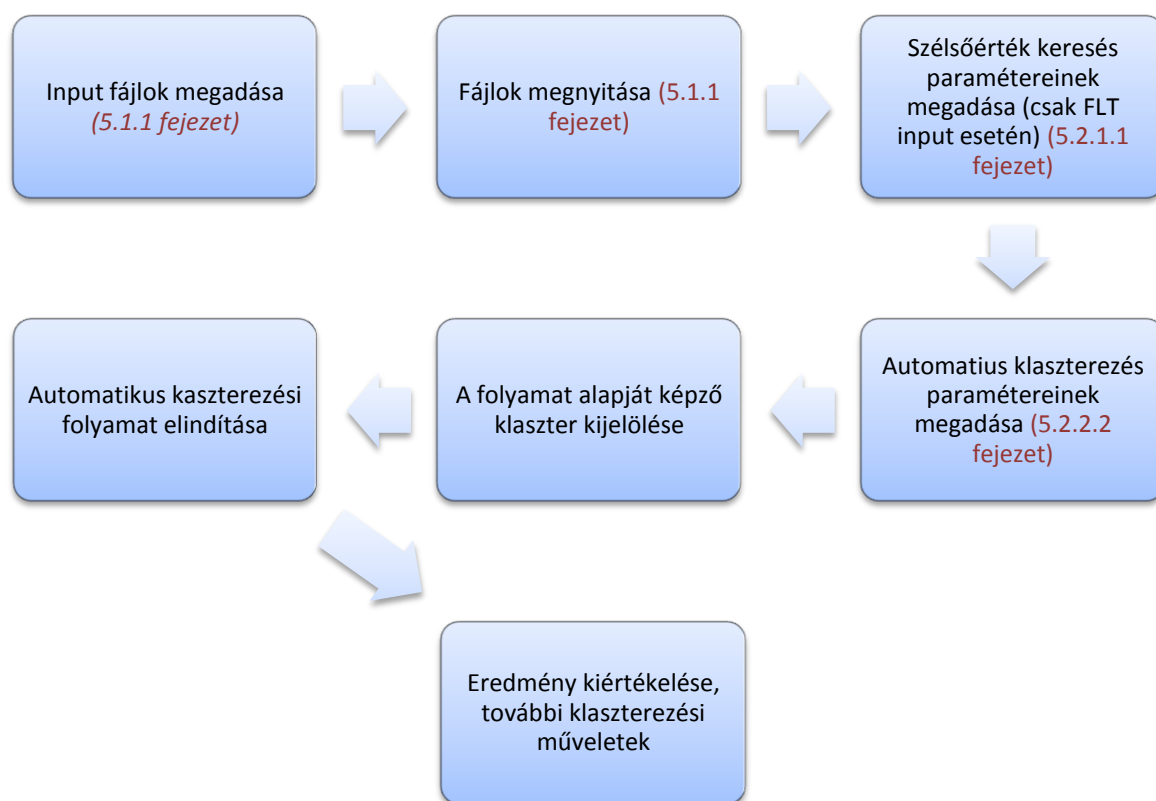


5.17. ábra: A kézi klaszterezés megvalósítása a pontfelhő megjelenítőkön

### 5.3.4 Az automatikus klaszterezés folyamata

Az automatikus klaszterezés során a program, beépített algoritmusai segítségével, a megadott paraméterek felhasználásával megpróbálja a hasonló formájú szívütéseket ugyanabba, az eltérő formájúakat különálló klaszterekbe szervezni.

A folyamat (5.18. ábra) alapját mindig egy klaszter (az aktuálisan kijelölt klaszter) képi.



5.18. ábra: Az automatikus klaszterezés folyamata

A művelet kimenete a kiindulási klaszterből előállított, szétválogatott QRS-ek, klaszterekbe csoportosítva. Az újonnan létrejött klaszterek a lista végére kerülnek, a legutolsó az úgynevezett *nem klaszterezhető QRS*-eket tartalmazza (amennyiben vannak ilyenek).

A nem klaszterezhető QRS-ek azok a QRS-ek, amelyek az automatikus szétválogatási folyamat során egyik csoportba sem voltak besorolhatóak.

### 5.3.5 Klaszterezési módszerek keverése

Az automatikus klaszterezés elsődlegesen a klaszterezési folyamat segítésére, gyorsítására használható. Önálló alkalmazása sok esetben nem ad kielégítő eredményt. A legjobb eredmény a kézi és az automatikus módszer keverésével érhető el.

Tetszőleges, kijelölt klaszteren végrehajthatók automatikus, valamint kézi szétválogatási műveletek is, azok sorrendjére, variációjára nincs megkötés. Egy tipikus forgatókönyv az,

amikor a bemeneti állományok megnyitása után a „durvább” szétválogatásra először az automatikus módszert, a finomabb klaszterek kidolgozására pedig a kézi módszert használjuk.

## 6. Összefoglalás

A Labtech Kft. által felvetett problémák megoldására alakult munkacsoportunk 2009 márciusában megkezdett, 2010 áprilisában befejezett tevékenysége után elmondható, hogy a cég által felállított kritériumokat csoportunk teljesítette, munkánkat a cég átvette.

Az alkalmazásunk demonstrációs- és tesztelési céllal jött létre, a program működését biztosító osztályok (a szűrést valamint az automatikus klaszterezést megvalósító osztályok, kézi klaszterezésért valamint a megjelenítésért felelős osztályok) az átadás után a Labtech Kft. kereskedelmi forgalomba kerülő, orvosi diagnosztikai szoftverébe épülhetnek be.

Az alkalmazáshoz egy fejlesztői dokumentáció is készült, mely a program mellett átadásra került.

Az létrehozott osztályokon a Labtech Kft. további finomításokat, teljesítményoptimalizálást valamint testre szabást hajtott végre, ezt követően a keretalkalmazásukba az alkalmazás által biztosított interfészekon keresztül beépítésre kerül.

A beépítés folyamata előrehaladott állapotban van. A béta tesztelések orvosok közreműködésével lezajlottak, a visszajelzések értékelése és további finomítások után az alkalmazás kiadásra kerülhet.

A kitűzött feladatok java részben megoldásra kerültek (felvételek kezelése, alapvonal szűrés az FLT állományokon, a felvétel QRS-einek megjelenítése, szélsőérték pontok megjelenítése, klaszterek kezelése, kézi klaszterezés, automatikus klaszterezés, valós idejű feldolgozás), viszont maradtak megoldandó problémák is. Jelenleg nem megoldott az egy-egy klaszteren belüli QRS-ek hasonlóságának kiszámítása (a klaszter jóságának mértéke), valamint az automatikus klaszterezés után létrejövő klaszterek hasonlóságuk alapján való automatikus összevonása.

Munkánk során rengeteg tapasztalatot szereztünk a csapatmunka, a megrendelővel való kommunikáció, a C# .NET 2.0 programozás, a WindowsForms programozása, a kép- és jelfeldolgozás, a klaszterezési algoritmusok, a teljesítményoptimalizálás valamint a dokumentáció készítés területén.

## **7. Köszönetnyilvánítás**

Ez úton szeretnénk megköszönni Dr. Juhász István Tanárúrnak valamint Vágner Anikó Tanárnőnek a munkánk segítségét, a hasznos tanácsokat valamint a mindig segítőkész hozzáállásukat.

Tanárúrnak külön köszönettel tartozunk szakdolgozatunk koordinálásáért, témavezetőként adott hasznos tanácsaiért!

Köszönettel tartozunk a Labtech Kft-nek a lehetőségért, valamint Farkas Lászlónak, aki a Kft. kapcsolattartójaként mindig segítőkészen állt a kérdéseink és problémáink elé!

## 8. Irodalomjegyzék

[1] Schikuta E.:

Grid clustering: An efficient hierarchical clustering method for very large data sets  
Proc. 13th Int. Conf. on Pattern Recognition, Vol 2, 1996.

[2] M. Ankerst, M. Breunig, H. Kriegel, J. Sander:

Optics: ordering points to identify the clustering structure.

In SIGMOD'99: Proceedings of the 1999 ACM SIGMOD International Conference on  
Management of Data, 1999.

[3] M. Anderberg:

Cluster Analysis for Applications.

Morgan Kaufmann Publishers, 1973.

[4] R. Iváncsy, A. Babos, Cs. Legány:

Analysis and Extensions of Popular Clustering Algorithms

Department of Automation and Applied Informatics and HAS-BUTE Control Research Group  
Budapest University of Technology and Economics

[5] Pau Micó, David Cuesta, and Daniel Novák:

Clustering Improvement for Electrocardiographic Signals

Department of Systems Informatics and Computers, Polytechnic School of Alcoi,

Plaza Ferràndiz i Carbonell 2, 03801 Alcoi, Spain

Department of Cybernetics, Czech Technical University in Prague, Czech Republic

[6] P. Berkhin:

Survey of clustering data mining techniques.

Technical report, Accrue Software, 2002.

[7] J. Han, M. Kamber:

Data Mining: Concepts and Techniques.

Morgan Kaufman Publisher, 1972

(magyararul: Adatbányászat - Konceptiók és technikák. Panem, 2004).

[8] L. Rónyai, G. Ivanyos, R. Szabó:  
Algoritmusok (Algorithms).  
Typotex, 1999.

[9] P. Rózsa:  
Lineáris algebra és alkalmazásai.  
Műszaki Könyvkiadó, 1991 (3. kiadás).

[10] .NET 2.0 Reference.  
Microsoft Corporation One Microsoft Way Redmond, WA 98052-7329 USA  
<http://msdn.microsoft.com/en-us/library/aa338210.aspx>