

# **Vuorovaikutteinen visualisointitekniikka biosignaalin analysointiin**

Tiina Vainio

Tampereen yliopisto  
Informaatiotieteiden yksikkö  
Vuorovaikutteinen teknologia  
Pro gradu -tutkielma  
Ohjaaja: Harri Siirtola  
Joulukuu 2014

Tampereen yliopisto

Informaatiotieteiden yksikkö

Vuorovaikutteinen teknologia

Tiina Vainio: Vuorovaikutteinen visualisointitekniikka biosignaalin analysointiin

Pro gradu -tutkielma, 46 sivua

Joulukuu 2014

---

Sydänsairauksista saadaan lisätietoa tutkimalla sydänsolujen kalsiumsignaalin häiriöitä. Tässä tutkielmassa esitellään kalsiumsignaalien analysointiin kehitetty sovellus, joka pohjautuu visuaalisen analytiikan keinoihin. Sovellus toimii selaimessa, ja se perustuu laskennalliseen analyysiin, jonka tulokset visualisoidaan. Laskennan parametrien muutokset heijastuvat reaaliaikaisesti visualisointiin. Lisäksi sovellus mahdollistaa lääkevas-  
teiden analysoinnin ja vertailun.

Avainsanat ja -sanonnat: biosignaalit, visuaalinen analytiikka.

## Sisällysluettelo

1. Johdanto.....	1
2. Ongelman kuvaus.....	3
2.1 Kalsiumsignaali.....	3
2.2 Tavoitteet.....	6
2.3 Aiemmat ratkaisut.....	6
3. Visuaalinen analytiikka.....	9
3.1 Visuaalisen analytiikan prosessi.....	9
3.2 Vuorovaikutus.....	11
3.2.1 Vuorovaikutus prosessina.....	12
3.2.2 Vuorovaikutustekniikat.....	13
3.3 Esitystavoista.....	17
3.3.1 Ajallinen data.....	18
3.3.2 Moniulotteinen data.....	19
3.3.3 Useat näkymät.....	21
4. Ratkaisun kuvaus.....	23
4.1 Regressiosuorien laskenta.....	23
4.2 Poikkeamien laskenta.....	24
4.3 Tekniikat.....	26
4.3.1 HTML.....	26
4.3.2 JavaScript.....	27
4.4 Näkymät.....	28
4.4.1 Yhteenvedonäkymä.....	29
4.4.2 Datanäkymä.....	30
4.4.3 Analyysinäkymä.....	30
4.5 Toiminnot ja vuorovaikutus.....	31
5. Pohdinta.....	35
5.1 Käytettävyydestä.....	35
5.2 Jatkokehitys.....	37
5.2.1 Poikkeamat.....	37
5.2.2 Yhteenvedonäkymä.....	39
5.2.3 Signaalien vertailu.....	39
5.3 Tavoitteiden täytyminen.....	40
6. Yhteenvedo.....	42
Viiteluettelo.....	43

## 1. Johdanto

Biosignaalit ovat ikkuna kehon tilaan ja toimintaan. Niitä voidaan hyödyntää monin eri tavoin. Sovellukset vaihtelevat uudenlaisista käyttöliittymistä (esim. aivokäyttöliittymä World of Warcraft -pelille [Kapeller et al., 2012]) sairauksien tutkimiseen ja diagnosointiin. Sydänsähkökäyrä (elektrokardiogrammi, EKG) on useimmille tuttu biosignaali sydämen toiminnan tutkimiseen, mutta sydäntä voidaan tutkia myös kalsiumsignaalien avulla. Kalsium on tärkeässä osassa sydämen sykkeessä, ja esimerkiksi rytmihäiriöt näkyvät myös kalsiumsignaalissa.

Erityisen mielenkiintoiseksi kalsiumsignaalien tutkiminen on tullut Shinya Yamakan tutkimusryhmän kehittämän kantasolumenetelmän myötä. Kantasolut ovat soluja, jotka voivat erilaistua kaikiksi eliön kudostyypeiksi. Yamanakan menetelmä mahdollistaa jo erilaistuneiden solujen palauttamisen kantasoluiksi. Saadut kantasolut voidaan erilaistaa uudelleen eri tyyppisiksi. Sydänsairauksien tutkimisen suhteen menetelmä on mullistava, koska sydänsoluista olisi vaikea ottaa näytettä. Nyt voidaan ottaa näyte esimerkiksi ihosta ja muuntaa solut sydänsoluiksi. Solut ovat eläviä ja sykkivät spontaanisti, joten niiden kalsiumsignalointia voidaan tutkia. Lisäksi tuotetut solut sisältävät juuri kyseisen yksilön perimän, joten voidaan tutkia tietyn henkilön sydämen toimintaa ja vaikkapa vastetta lääkitykseen. Tämä helpottaa myös perinnöllisten sydänsairauksien tutkimista.

Kalsiumsignaalien epäsäännöllisyyksien tiedetään liittyvän erilaisiin sydänsairauksiin. Asiantuntijat pystyvät tunnistamaan, onko signaali normaali vai epänormaali, ja he ovat luokitelleet signaaleja vain katsomalla signaalien käyrää. Tästä johtuen signaalien erilaisia poikkeamatyyppejä ei ole vielä tutkittu kovin paljon, sillä yksityiskohtainen poikkeamien tunnistaminen ilman apuvälineitä on hidasta.

Kalsiumsignaalien luokittelua kannattaa siis nopeuttaa ottamalla tietokoneet apuun. Tässä tutkielmassa on kehitetty uudenlainen kalsiumsignaalien poikkeamien luokitteluun kehitetty sovellus. Poikkeamien luokittelu puhtaasti laskennallisilla keinoilla on hyvin haastavaa, joten ratkaisua lähestyttiin visuaalisen analytiikan keinoin. Visuaalinen analytiikka (visual analytics) yhdistää erilaisia tietokoneen ja ihmisen tiedonkäsittelyä ja vuorovaikutusta tutkivia aloja. Tarkoituksena on yhdistää ihmisen ja tietokoneen vahvuudet ongelmanratkaisussa.

Kehitetty sovellus on selainpohjainen vuorovaikutteinen visualisointi. Ohjelmaan kuuluvat poikkeamat tunnistavat algoritmit, joiden parametreja käyttäjä voi säätää. Parametriarvojen muutokset heijastuvat reaaliaikaisesti visualisointiin, joten käyttäjän on helppo hakea oikeat arvot parametreille.

Tutkielman rakenne on seuraava: Luvussa 2 kerrotaan tarkemmin kalsiumsignaalista ja sen luokittelusta sekä poikkeavuuksista. Luvussa kuvataan myös aiheeseen liittyviä

aiempia ratkaisuja sekä asetetut vaatimukset tässä työssä kehitellylle ja kuvatulle ratkaisulle.

Luvussa 3 on yleistä taustaa visuaalisesta analytiikasta ja siihen liittyvästä prosessista sekä vuorovaikutuksen merkityksestä. Lisäksi luvussa käsitellään visualisointitekniikoita, kuten toteutettavaksi valitussa ratkaisussa hyödynnettyjä useita näkymiä.

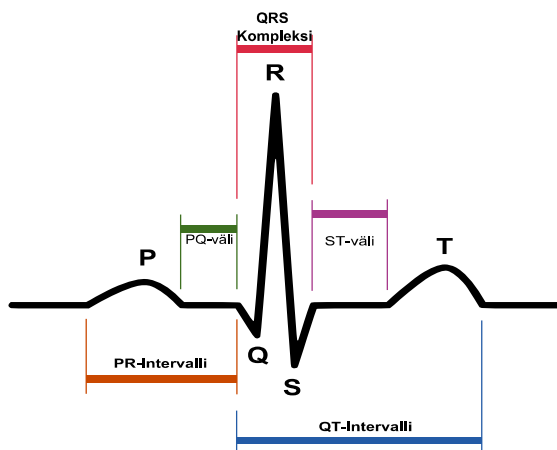
Luvussa 4 kuvataan ratkaisu eli kehitetty vuorovaikutteinen visualisointisovellus. Luvussa kerrotaan, miten poikkeamat lasketaan ja millaisia parametreja laskennassa käytetään. Siinä esitellään myös sovelluksen näkymät, niiden väliset yhteydet ja käyttäjän ja sovelluksen välinen vuorovaikutus. Luvussa myös käsitellään ohjelman toteutuksessa käytettyjä tekniikoita.

Luvussa 5 käsitellään ohjelman nykyversion käytettävyyttä ja mahdollisia puutteita, arvioidaan luvussa 2 kuvattavien tavoitteiden täyttymistä sekä pohditaan tiedettyjä ja arvioituja jatkokehitystarpeita. Lopuksi luvussa 6 tehdään yhteenveto käsitellyistä asioista.

## 2. Ongelman kuvaus

Biosignaaleilla tarkoitetaan erilaisia kehosta mitattavia (jatkuvia) suureita. Monille tuttuja biosignaaleja ovat esimerkiksi elektrokardiogrammi EKG (Kuva 1), joka kuvaa sydämen sähköistä toimintaa, ja elektroenkefalogrammi EEG eli aivosähkökäyrä. Biosignaalit ovat tärkeitä lääketieteessä, sekä sairauksien ja lääkityksen tutkimisessa että yksittäisten potilaiden diagnosoinnissa, sillä niitä tutkimalla saadaan selville erilaisia asioita kehon toiminnasta ja tilasta.

Jotta signaaleista saa tietoa, niistä täytyy pystyä tekemään havaintoja: löytää toistuvia tai poikkeavia hahmoja (pattern) tai muita lainalaisuuksia. Avuksi on kehitetty erilaisia tietokonesovelluksia. Eri signaaleihin liittyy erilaisia analysointitehtäviä ja ongelmia. Tässä tutkielmassa kuvataan ohjelmistoratkaisu yhden biosignaalin, kalsiumsignaalin, analysoinnin avuksi. Ratkaisussa keskitytään sydänsolujen kalsiumsignaalien poikkeamien tunnistamiseen ja signaalien luokitteluun normaaleiksi tai epänormaaleiksi.



Kuva 1: EKG-signaali

### 2.1 Kalsiumsignaali

Kalsiumsignaali on biosignaali, joka kuvaa kalsiumionien pitoisuuden vaihtelua soluli-massa. Kalsium on solujen toisolähetti eli se välittää viestejä solukalvolta solun sisään. Se on tärkeässä osassa lihassolun supistumisessa, joten se on merkittävä myös sydämen toiminnan kannalta. Sydänsoluissa kalsium vaikuttaa muun muassa sähköiseen toimintaan ja sydämen sykkeeseen sekä energian kulutukseen ja tuottamiseen [Bers, 2008].

Solua ympäröivällä solukalvolla on jännite, joka johtuu erilaisista ionien eli varautu-neiden hiukkausten pitoisuuksista solun sisä- ja ulkopuolella. Solukalvossa on erilaisia kanavia, joiden kautta ionit pääsevät kulkemaan solun sisään ja sieltä ulos. Nämä kana-vat aukeavat tietyillä jännitetasoilla – eri tyyppiset kanavat eri tasoilla. Kanavien aukea-mien aiheuttaa ionivirtauksia, jotka muuttavat taas solukalvon jännitetasoa. Lisäksi io-nien liikkumiseen vaikuttaa aktiiviset mekanismit, jotka siirtävät tiettyjä ioneita solusta

sisään tai ulos. Ionivirtoja on vastaavasti myös solun sisäisten soluelinten ja soluliman välillä.

Nämä ionivirrat aiheuttavat toimintapotentiaalin – esimerkiksi hermosolun aktivoitumisen tai lihassolun supistumisen. Normaalisti solukalvolla vallitsee tietty jännitetaso, lepopotentiaali. Aktiopotentiaalissa jännite nousee (depolarisoituu) ja taas laskee (repolarisoituu) nopeasti. Normaalisti nämä tapahtumat vaativat ulkoisen ärsyksen, mutta sydämessä osa soluista on tahdistinsoluja, jotka depolarisoituvat spontaanisti aiheuttaen sydämen syke.

Kalsiumionit vaikuttavat solukalvon jännitetasoon ja sen muutoksiin. Kalsiumionivirrat vaikuttavat sekä tahdistinsolujen depolarisaatioon että muiden sydänsolujen supistumiseen ja palautumiseen. Häiriöt kalsiumaineenvaihdunnassa voivat siten aiheuttaa rytmihäiriöitä. [Bers, 2008]

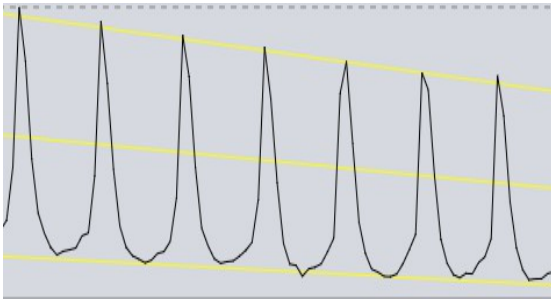
Normaalisti sydänsolujen kalsiumpitoisuus vaihtelee säännöllisesti, ja yksi signaalin piikki vastaa yhtä sydämenlyöntiä. Kuvassa 2 on esitetty normaali kalsiumsignaali. Koska tämä signaali vaikuttaa oleellisesti sydämen sykkeeseen, näkyvät esimerkiksi rytmihäiriöt signaalin muodossa. Kalsiumsignaalin poikkeavat muodot liittyvätkin eri sydänsairauksiin. Signaalista voidaan tunnistaa eri tavoin poikkeavia muotoja.

Tässä työssä on tunnistetty kuusi poikkeamatyyppiä: matala piikki (low peak), keskipiikki (middle peak), monihuippuinen piikki (multiple peak), alapiikki (bottom peak), tasopoikkeama (platform abnormality) ja epäsäännöllinen vaihe (irregular phase). Matala piikki tarkoittaa pientä ylöspäin suuntautuvaa piikkiä aallonpohjassa (Kuva 3). Keski-piikki on vastaava pieni piikki matkalla aallonpohjalta aallonhuipulle tai päinvastoin (Kuva 4). Monihuippuinen piikki viittaa useampaan aallonhuippuun, joiden välissä ei ole tarpeeksi syvää aallonpohjaa (Kuva 5). Alapiikki on aallonpohja, joka on normaalia syvempi (Kuva 6). Tasopoikkeama merkitsee hetkellistä nousu- tai laskunopeuden muutosta (Kuva 7). Epäsäännöllinen vaihe tarkoittaa poikkeavan lyhyttä tai pitkää kahden huipun väliä (Kuva 8).

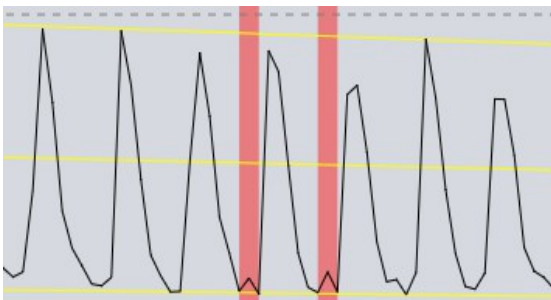
Vaikka erilaisia poikkeamia on pystytty tunnistamaan ihmissilmin, signaaleja on aiemmin luokiteltu vain joko normaaliksi tai epänormaaliksi. Tämä johtuu siitä, että asiantuntijat ovat tehneet luokittelua ilman apuvälineitä ainoastaan katsomalla signaalia, ja tällainen manuaalinen luokittelu yksityiskohtaisemmalla tasolla veisi liikaa aikaa.

Poikkeamien tarkempi tutkiminen on kuitenkin kiinnostavaa eri sydänsairauksien patologian tutkimisessa: voidaan esimerkiksi selvittää liittyvätkö tiettyntyyppiset poikkeamat, niiden yhdistelmät tai poikkeamien puuttuminen tiettyihin sairauksiin.

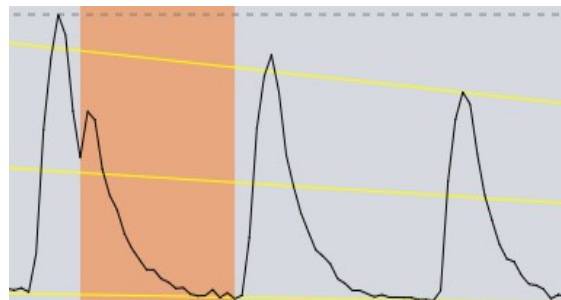
Poikkeamien manuaalinen luokittelu on ongelmallista paitsi ajankäytöllisesti, myös yhdenmukaisuuden suhteen. Kun luokittelu on subjektiivista, eri asiantuntijat voivat tulla erilaisiin johtopäätöksiin, ja jopa sama henkilö voi eri päivinä luokitella signaalin eri tavoin.



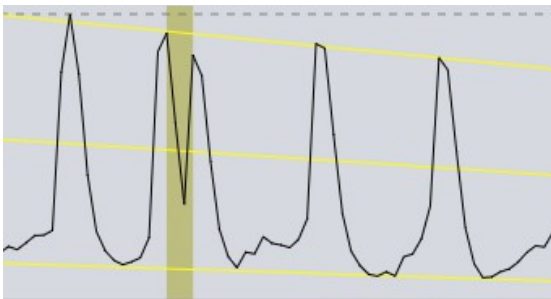
*Kuva 2: Normaali kalsiumsignaali*



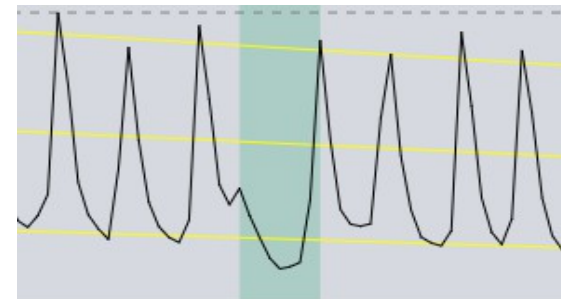
*Kuva 3: Matala piikki*



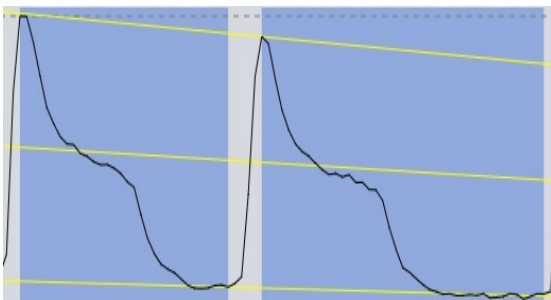
*Kuva 4: Keskipiikki*



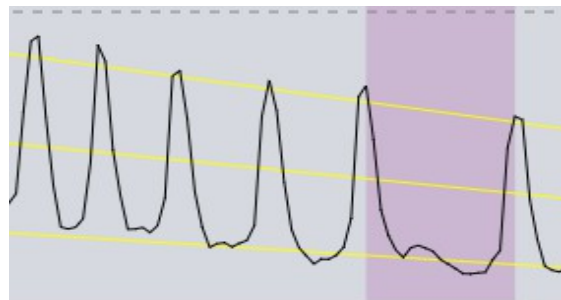
*Kuva 5: Monihuippuinen piikki*



*Kuva 6: Alapiikki*



*Kuva 7: Tasopoikkeama*



*Kuva 8: Epäsäännöllinen vaihe*



Kalsiumsignaaleja ja niiden poikkeamia käytetään myös tutkittaessa lääkevastetta eli kuinka sydän reagoi lääkkeeseen. Tällöin vertaillaan signaalia ennen ja jälkeen lääkityksen ja voidaan nähdä, onko lääke poistanut poikkeamat tai aiheuttanut muita vaikutuksia signaaliin.

Kalsium on siis merkittävä sydämen toiminnan kannalta, ja sen pitoisuuden vaihtelua mittaamalla saadaan signaali, jonka avulla sydämen toimintaa voidaan tutkia.

## 2.2 Tavoitteet

Tässä tutkielmassa esiteltävän ratkaisun tavoitteena oli ensinnäkin nopeuttaa normaali/epänormaali-luokittelua automatisoimalla sitä. Täysin automaattinen menetelmä on kuitenkin haastava kehittää, kuten asiaa tutkineet Juhola ja muut [2004] toteavat. Lähestymistavaksi päätettiin siten valita vuorovaikutteinen visualisointi, joka tunnistaa poikkeamat ja jonka tunnistusparametreja voidaan dynaamisesti säätää.

Toisekseen haluttiin mahdollistaa yksityiskohtaisempi luokittelu eli poikkeamien tunnistaminen, joka mahdollistaisi signaalien tarkemman tutkimisen hidasta manuaalista luokittelua tehokkaammin. Lisäksi menetelmän tuli olla objektiivinen, jotta sen avulla voitaisiin tehdä kalsiumsignaalin luokitteluun liittyviä tutkimuksia, jotka olisivat muiden tutkijoiden toistettavissa. Objektiivisuus on luonnollisesti seurauksena, kun laskentatapa täytyy määritellä yksityiskohtaisesti ohjelmoitavaa toteutusta varten.

Kolmas tarvittu toiminto luokittelun ja poikkeamien tunnistamisen lisäksi oli signaalien vertailu, jotta tutkijat voisivat nähdä lääkkeiden vaikutuksen signaaliin. Tämä toiminnallisuus on jokseenkin irrallinen – se ei liity signaalin käsittelyyn ja laskentaan mitenkään, mutta koska tarpeena oli nähdä muutokset ohjelman tekemässä luokittelussa ja poikkeamissa, oli toiminnon liittäminen ohjelmaan perusteltua.

Lisäksi teknisinä vaatimuksina oli alustariippumattomuus ja se, ettei ohjelma vaatisi asentamista, koska käyttäjillä ei välttämättä aina ole asennusoikeuksia päätteelleen. Erityisesti menetelmän kehitysvaiheessa, jolloin käyttäjät voivat saada useinkin päivitettyjä versioita ohjelmasta, olisi hankalaa, jos aina tarvittaisiin IT-tuen apua.

Tavoitteena oli siis luoda vuorovaikutteinen visualisointi, joka nopeuttaisi ja tarkentaisi signaalien luokittelua, toisi luokitteluun objektiivisuutta, ja joka olisi helppo ottaa käyttöön eri laitteilla tarvitsematta asentaa mitään. Viimeinen vaatimus johti selainpohjaisen ratkaisun valitsemiseen.

## 2.3 Aiemmat ratkaisut

Koska erilaiset signaalit ovat tärkeitä lääketieteessä, on niiden käsittelyä tutkittu paljon ja niiden analysointiin on kehitetty runsaasti laskennallisia menetelmiä. Myös signaalien automaattista luokittelua on tutkittu [Wong et al., 2014]. Tyypillisesti menetelmät koskevat tiettyä signaalityyppiä, eivätkä yleisty muihin, ainakaan sellaisenaan. Kuhunkin

signaaliin kun liittyy omanlaisensa normaali käyrän muoto ja sen poikkeamat. Kuvista 1 ja 2 näkyy, kuinka erilaisia esimerkiksi sydänsähkökäyrä ja kalsiumkäyrä ovat.

Suosituimpia tutkimuskohteita ovat EKG- ja EEG-signaalit, mutta myös kalsiumsignaalin luokittelun automatisoimista on yritetty. Kalsiumsignaali liittyy muihinkin kuin vain sydänsoluihin, ja sen luokittelua on pyritty automatisoimaan ainakin T-lymfosyyttien yhteydessä [Payne et al., 2005]. Luokat ovat erilaiset ja signaalitkin poikkeavat kohdassa 2.1 esitellystä. Toisin kuin sydänsoluissa, lymfosyyttien kohdalla ei ole kyse jatkuvasti toistuvasta ärsykkeestä, joka saa kalsiumin vapautumaan, vaan siinä on analysoitu yksittäistä ärsykettä ja sen vastetta.

Sydänsolujen kalsiumsignaalin luokitteluun on kehitetty ilmeisesti vain yksi laskennallinen menetelmä [Juhola et al., 2014]. Tällä menetelmällä kuitenkin pyrittiin ainoastaan erottelamaan normaalit ja epänormaalit signaalit toisistaan. Luokittelussa testattiin eri tapoja: k lähintä naapurua (K-nearest neighbor), diskriminanttianalyysiä (discriminant analysis), naiivia bayesiläistä menetelmää (naïve Bayes rule) ja luokittelupuita (classification trees). Näitä verrattiin asiantuntijan suorittamaan manuaaliseen luokitteluun.

Automatisoidun luokittelun tarkkuus vaihteli käytetystä signaalijoukosta ja luokittelumenetelmästä riippuen noin 70-90%. Tarkkuuden suhteen parhaimmissa tuloksissa tosin spesifisyys eli oikeiden negatiivisten tulosten tunnistaminen oli heikkoa. Juhola ja muut toteavatkin käytetyn datan automaattisen luokittelun olevan vaikeaa, koska signaalit ovat lyhyitä ja niissä on vain vähän huippuja.

Biosignaalien tutkimiseen on käytetty myös visuaalista lähestymistapaa, ja niille on kehitetty selainpohjaisia visualisointityökaluja aiemminkin, esimerkiksi Cloudwave EEG:n tallentamista ja analysointia varten [Jayapandiana et al., 2013]. Kyse on kuitenkin aivan erilaisesta tehtävästä ja toiminnallisuudesta kuin kalsiumsignaalin tapauksessa. Cloudwave on tarkoitettu suurille datamäärille, sillä EEG on monikanavaista dataa ja lisäksi tutkimukseen liittyvät mittaukset pitkiä. Cloudwaven tarkoituksena onkin löytää datasta merkitseviä osioita ja myös yhdistää EEG-signaalidataa muihin mittaustietoihin.

Lourenço ja muut [2014] puolestaan ovat kehittäneet SignalBITin, eri signaalityypeille käyvän www-pohjaisen alustan biosignaalien visualisointiin ja merkintöjen tekemiseen. Tässä työkalussa on hyödynnetty useita näkymiä, ja siinä pystyy tutkimaan ja vertaamaan useita signaaleja kerrallaan. Tämä sovellus ei kuitenkaan sisällä laskentaa, vaan signaalien eri kohtien merkinnät jäävät asiantuntijoiden tehtäväksi.

Biosignaalien luokitteluun kehitetyt visuaaliset työkalut eivät juurikaan hyödynnä laskennallisuutta, siinä missä tiedonlouhinnan menetelmiä käytettäessä ei hyödynnetä visuaalisia apukeinoja. Kalsiumsignaalien luokitteluun on kehitelty myös ratkaisua, joka yhdistää visuaalisia ja laskennallisia keinoja: AnomalyExplorer [Siirtola et al., 2014].

AnomalyExplorer on Java-ohjelmointikielillä toteutettu työkalu, jonka jatkokehitykseen tässä esiteltävä ratkaisu perustuu.

### 3. Visuaalinen analytiikka

Visuaalinen analytiikka (visual analytics) on noussut tutkimuksen kohteeksi 2000-luvulla datamäärien kasvaessa valtaviksi tietokoneiden tehojen ja tiedonsiirtonopeuksien jatkuvan kasvun ja halpenemisen seurauksena. Tarkoituksena on yhdistää ihmisen intuitio ja matemaattiset menetelmät hahmojen tunnistamiseksi ja tietämyksen ja oivalluksen tuottamiseksi [Wong and Thomas, 2004]. Visuaalinen analytiikka on analyttistä päätelyä vuorovaikutteisten, visuaalisten käyttöliittymien avustamana [Thomas and Cook, 2005].

Visuaalinen analytiikka yhdistää useita aloja, kuten tiedonhallintaa, -louhintaa, tiedon visualisointia ja kognitiotiedettä. Visuaalinen analytiikka on hyödyksi silloin, kun ongelma on liian epämääräinen ratkaisun (täydelliseen) automatisoimiseen ja kuitenkin sen verran monimutkainen, että se on liian työläs ihmiselle ilman apuvälineitä.

Visualisointi on tehokas tapa välittää tietoa ihmiselle: näköaistin kautta ihminen saa enemmän tietoa kuin kaikkien muiden aistien kautta yhteensä [Ware, 2012, 2]. Niinpä visualisointi mahdollistaa suuren datamäärän omaksumisen. Toisaalta visualisointi auttaa huomaamaan erilaisia poikkeavuuksia ja hahmoja, mikä avustaa hypoteesien muodostamisessa datan pohjalta.

Hyvin yksinkertainen esimerkki visuaalisen esitystavan merkityksestä on kuvassa 9, vaikkei siinä varsinaisesta visualisoinnista olekaan kyse. Jos lukulistasta täytyy etsiä suurin luku, vasemmanpuoleinen lista täytyy lukea läpi pitäen mielessä, mikä on siihen mennessä löytynyt suurin luku. Kun desimaalipilkut on kohdistettu oikeanpuoleisessa listassa, näkee heti, että suurin luku on se, jossa on eniten numeroita pilkun vasemmalla puolella. Lukuja ei tarvitse vaivautua tarkemmin prosessoimaan.

10,8	10,8
32,476	32,476
1088,9	1008,9
45,459	45,459
1,59	1,59
89,011	89,011

*Kuva 9: Yksinkertainen esimerkki visuaalisen esitystavan hyödystä: listan suurin luku on helppo nähdä, kun desimaaliluvut on kohdistettu*

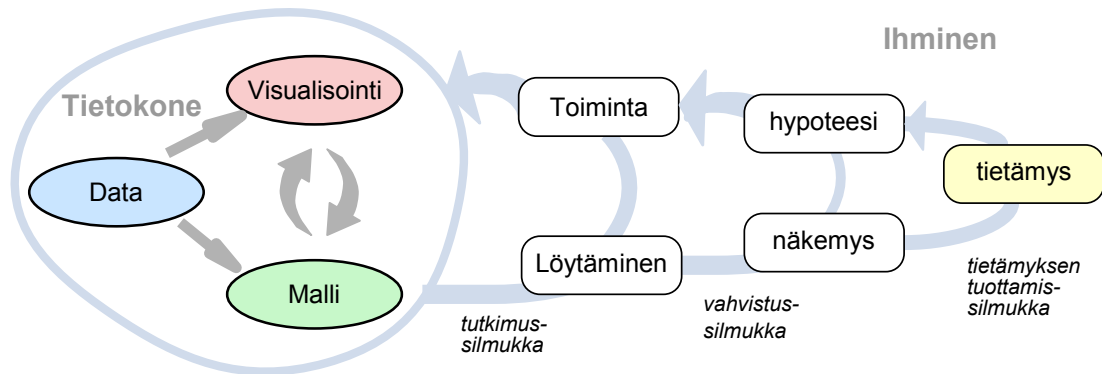
#### 3.1 Visuaalisen analytiikan prosessi

Visuaalisen analytiikan sovellusten perimmäisenä tarkoituksena on tukea tietämyksen muodostumista päätöksentekoa varten. Tietämys rakentuu tiedon varaan, joka puolestaan perustuu dataan. Datalla itsellään ei ole merkityksiä, se on järjestämätöntä ja prosessoimatonta, mutta siitä tulee tietoa järjestämisen ja prosessoinnin kautta; tietämyksessä taas yhdistyvät tieto, ymmärrys, kokemus, taidot ja arvot [Rowley, 2007]. Tietä-

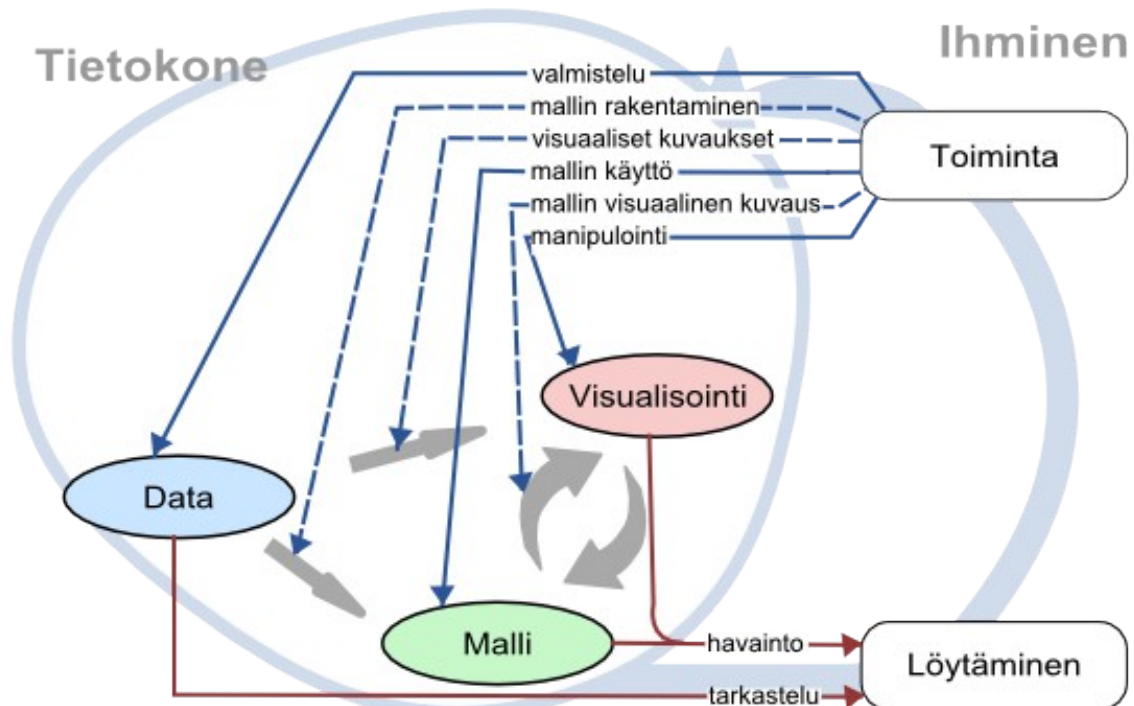
myksen lisäksi visualisoinnin yhteydessä usein puhutaan oivalluksesta tai näkemyksestä (insight). Vaikka englanninkielisessä kirjallisuudessa näistä käytetäänkin samaa termiä insight, eri kirjoittajilla merkitys selvästi vaihtelee ahaa-elämyksestä kuvan muodostamiseen datasta.

Tietämyksen muodostuminen on prosessi, ja sille on esitetty visuaalisessa analytiikassa useampiakin malleja. Näistä ainakin Kerrenin ja Schreiberin [2012] sekä Sachan ja muiden [2014] versiot huomioivat ihmisen ja tietokoneen välisen työnjaon. Malleissa on paljon yhteistä, ja tässä kuvataan tarkemmin niistä jälkimmäinen.

Sachan ja muiden [2014] malli on esitetty on kuvissa 10 ja 11. Malli on jaettu kahteen osaan: tietokoneeseen ja ihmiseen, pyrkiihän visuaalinen analytiikka yhdistämään näiden vahvuudet. Malli koostuu kolmesta silmukasta: tutkimus (exploration), vahvistus (verification) ja tietämyksen tuottaminen (knowledge generation loop).



Kuva 10: Tietämyksen muodostumisen malli [Sacha et al., 2014]



Kuva 11: Yksityiskohdat toimintaan ja kognitioon liittyen [Sacha et al., 2014]

Tutkimussilmukka kuvaa analysoijan vuorovaikutusta järjestelmän kanssa käyttökelpoisten löytöjen tekemiseksi (*toiminta ja löytäminen* kuvassa 10). Analysoija voi vaikuttaa dataan ja sen käsittelyyn, siihen miten datasta rakennetaan malli sekä kuinka data tai malli visualisoidaan. Analysoija voi siten löytää eli tehdä kiinnostavia huomioita datasta. Huomiot puolestaan johtavat uusiin toimintoihin, vaikkapa visuaalisen kohteen zoomaamiseen.

Vahvistussilmukka ohjaa tutkimussilmukkaa hypoteesien vahvistamiseksi tai muodostamiseksi. Tutkimussilmukan löydökset johtavat oivallukseen tai näkemykseen, joka vaikuttaa hypoteesien muodostamiseen. Hypoteesien muodostamiseen vaikuttaa myös analysoijan aiempi tietämys. Tätä kautta aiempi tietämys vaikuttaa koko uuden tietämyksen tuottamiseen: se ohjaa analyysistrategiaa ja prosessia. Vahvistussilmukan tuottamat näkemykset johtavat uuteen tietämykseen ja näin muodostuu tietämyksen tuottamissilmukka.

Tietämyksen tuottamisprosessissa kaikki alkaa datasta, ja analysoija voi valita vapaasti joko datan visualisoinnin tai mallin muodostamisen tai jonkinlaisen yhdistelmän näistä. Tavoitteena on päätyä uuteen tietämykseen, joka saavutetaan kuvattujen silmukoiden avulla.

### 3.2 Vuorovaikutus

Koska visuaalisen analytiikan tarkoituksena on yhdistää ihmisen ja tietokoneen vahvuudet, vuorovaikutus on oleellinen osa visuaalisen analytiikan järjestelmiä. Tyypillisesti vuorovaikutus käsitetään ihmisen ja järjestelmän väliseksi, mutta Pike ja muut [2009] esittävät, että visuaalisessa analytiikassa on vuorovaikutusta myös ihmisen ja ongelma-avaruuden välillä.

Vuorovaikutuksen määrittely on vaikeaa ja sitä onkin lähestytty eri tavoin. Vuorovaikutus on jatkuva prosessi, mutta usein on käytännöllisempää jakaa se pieniin osiin. Tällöin puhutaan vuorovaikutustekniikoista, tehtävistä tai toiminnasta. Eri termit korostavat erilaisia näkökulmia: tehtävä on jotakin, mitä ihminen suorittaa; toiminta on jotakin, minkä järjestelmä mahdollistaa.

Jansen ja Dragicevic [2013] näkevät vuorovaikutuksen käyttäjän mahdollisuutena muuttaa tiedon visualisoinnin ”jatkumoa”, joka kuvaa erilaiset vaiheet, joiden kautta raakadatasta saadaan visualisointi (Kuva 12). Lee ja muut [2012] lähestyvät asiaa ajallisuuden kautta: vuorovaikutuksella on alku ja loppu. Se jakautuu neljään mahdolliseen vaiheeseen: käyttäjän aikomus, käyttäjän toiminta, systeemin vaste ja palaute. Yi ja muut [2007] määrittelevät vuorovaikutustekniikat visualisoinnissa ”piirteinä, jotka tarjoavat käyttäjälle tavat joko suoraan tai epäsuoraan manipuloida ja tulkita esitystapoja”.

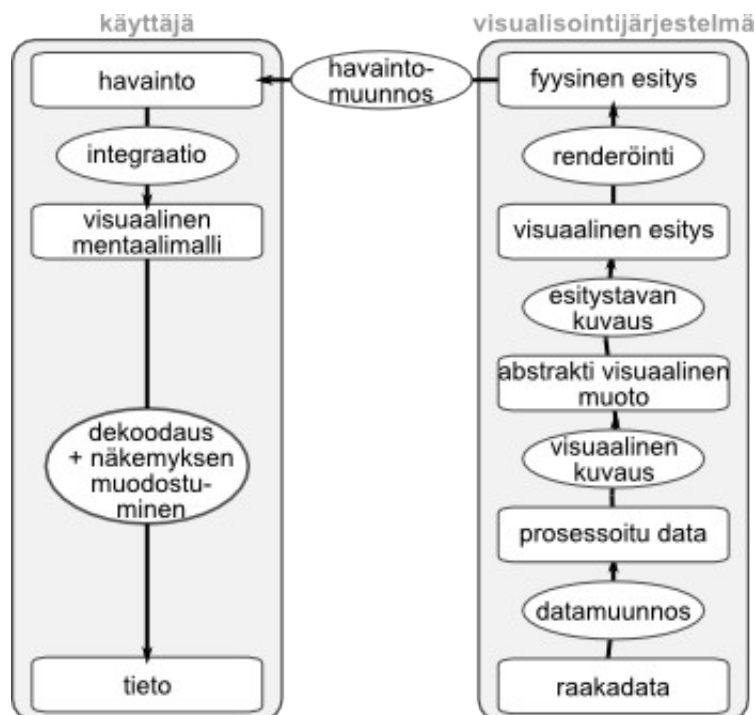
Edellä esitetyssä visuaalisen analytiikan prosessimallissa [Sacha et al., 2014] ihmisen ja tietokoneen rajapinta on tutkimussilmukassa, johon liittyy toiminta ja löytäminen

(Kuva 11). Toiminnot ovat ”yksittäisiä tehtäviä, jotka tuottavat käsin kosketeltavia, ainutkertaisia vasteita visuaalisen analytiikan järjestelmältä”. Nämä tehtävät voivat liittyä datan valmisteluun, mallin rakentamiseen, datan visuaalisten vastaavuuksien määrittelyyn, mallin käyttöön, mallin visuaalisten vastaavuuksien määrittelyyn tai visualisoinnin manipulointiin. Toiminnot johtavat havaintojen ja tarkastelun kautta löytämiseen.

### 3.2.1 Vuorovaikutus prosessina

Sachan ja muiden [2014] malli voidaankin nähdä kokonaisuutenakin vuorovaikutusmallina, vaikka ohjelmiston varsinainen käyttö rajautuu tutkimussilmukkaan. Käyttöä kuitenkin ohjaavat korkeamman tason silmukat eli käyttäjän tavoitteet ja pyrkimys tuottaa uutta tietämystä.

Vuorovaikutuksen eri prosessimallit tuovat esiin hieman erilaisia vaihteita ja näkökulmia. Ware [2012, 345] kuvaa vuorovaikutuksen visualisoinnin kanssa jakautuvan kolmeen lomittuvaan silmukkaan. Nämä silmukat ovat datan manipulointisilmukka, tutkimus- ja navigointisilmukka sekä ongelmanratkaisusilmukka. Tässä yhteydessä data viittaa visuaalisiin objekteihin eikä niinkään dataan, jota objektit esittävät. Datan manipulointi tarkoittaa näiden objektien valintaa ja liikuttamista silmän ja käden koordinaatiolla. Tutkimus- ja navigointisilmukassa käyttäjä ”suunnistaa” visualisoinnissa. Ongelmanratkaisusilmukassa käyttäjä hyödyntää visualisointia hypoteesien muodostamiseen. Silmukoista kaksi ensimmäistä liittyvät selvästi Sachan ja muiden mallin tutkimussilmukkaan, kun taas viimeinen vahvistussilmukkaan ja tietämyksen tuottamiseen.



Kuva 12: Jansen ja Dragicevicin [2013] versio tiedon visualisoinnin "jatkumosta"

Gotz ja Zhou [2008] tutkivat käyttäjien toimia heidän suorittaessaan analyysitehtäviä. He havaitsivat käyttäjien lähestyvän tehtäviään hierarkkisesti eli purkavan korkean tason tehtävät pienempiin ja konkreettisempiin. Lisäksi se huomasivat prosessin jakautuvan kahteen vaiheeseen: tutkimus ja oivallus. Tutkimusvaiheessa esimerkiksi suodataan dataa ja muutetaan näkymiä. Oivallusvaiheessa dokumentoidaan tehdyt löydökset: tehdään muistiinpanoja tai tallennetaan visualisoinnista kuva.

Kaikki kolme mallia kuvaavat samaa asiaa hieman eri jaottelulla. Sekä Sachalla ja muilla että Gotzilla ja Zhoulla on matalimmalla tasolla tutkimussilmukka, jonka Ware puolestaan on jakanut kahteen: datan manipulointiin sekä tutkimukseen ja navigointiin korostaen myös fyysisen vuorovaikutuksen merkitystä. Näitä korkeammalla tasolla ollaan ihmisen mielessä: Gotzin ja Zhoun oivalluksessa tai Waren ongelmanratkaisusilmukassa. Sachan ja muiden voi joko ajatella jakaneen tuon vaiheen kahteen tai ottaneet mukaan vielä korkeamman tason eli tietämyksen.

Prosessimallissa voi nähdä myös Piken ja muiden [2009] ajatuksen vuorovaikutuksesta paitsi järjestelmän kanssa, myös ongelma-avaruuden kanssa. Jälkimmäinen on läsnä prosessimallien korkeamman tason silmukoissa: puhutaanpa sitten ongelmanratkaisusta tai oivalluksen tai tietämyksen tavoittelusta, kyseessä on ihmisen mielessään muodostama käsitys datasta (tai maailmasta datan perusteella), ja tietokone on vain apuväline.

### 3.2.2 Vuorovaikutustekniikat

Vuorovaikutustekniikoita tai käyttäjän suorittamia tehtäviä on luokiteltu useista eri lähtökohdista. On matalan tason luokitteluja, jotka vastaavat kysymykseen *miten* tehtävä suoritetaan, ja korkean tason luokitteluja, jotka kertovat *miksi* tehtävä suoritetaan [Brehmer and Munzner, 2013]. Matalan tason mallit voi nähdä myös *systemikeskeisenä* ja korkean tason *ihmiskeskeisenä* [Yi et al., 2007]. Eri lähestymistavoilla on puolensa: kun luokitellaan käyttäjän tarkoituksen (intent) perusteella systeemin tarjoamien toimintojen sijasta, saadaan koottua lukuisat matalan tason tekniikat muutamaan korkeamman tason kategoriaan [Yi et al., 2007]. Lisäksi käyttäjän tarkoitus on otettava käyttöliittymäsuunnittelussa lähtökohdaksi, visualisoinnin tarkoituksena on tukea käyttäjää ymmärtämään informaatiota [Pike et al., 2009]. Toisaalta toiminnot ovat mitattavissa esimerkiksi hiiren klikkauksina, toisin kuin käyttäjän tarkoitukset [von Landesberger, 2014]. Sitä tarvitaan, kun toteutetaan erilaisia historiatoimintoja, jotka vaihtelevat yksinkertaisesta kumoamisesta jopa visuaaliseen esitykseen käytön eri vaiheista.

Käyttäjän tarkoitus ei ole mitenkään yksiselitteinen käsite. Sekä Yi ja muut [2007] että Gotz ja Zhou [2008] ottavat tarkoitukset lähtökohdaksi, mutta lopputulos on hyvin erilainen. Gotzin ja Zhoun tarkoitukset ovat datan muutokset, visuaaliset muutokset, muistiinpanojen muutokset ja historian muutokset. Toisaalta Gotz ja Zhou myös ryhmittelevät toiminnot tutkimukseen, oivallukseen ja metatoimintoihin. Tämä luokittelu on



esitetty kuvassa 13. Yi ja muut näkevät tarkoitukset esimerkiksi jonkin kiinnostavan asian merkitsemisenä tai haluna nähdä asiat eri tavalla järjestettynä. Kaikki tarkoitukset ja niihin liittyvät kategoriat on esitetty taulukossa 1. Brehmer ja Munzner [2013] ovat ottaneet oman mallinsa lähtökohdaksi kysymykset *mitä*, *miksi* ja *miten*. Miksi-kysymyksen voisi ajatella kuvaavan käyttäjän tarkoituksia. Käyttäjä voi joko *tuottaa* (muuntaa tai johtaa dataa tai tehdä merkintöjä) tai *kuluttaa* (kommunikoida tietoa muille, muodostaa tai vahvistaa hypoteeseja sekä muuten vain tutkailla visualisointia). Kuluttaminen johtaa mielenkiintoisten elementtien *etsimiseen* ja kun sellaisia on löytynyt, tehdään *kyselyjä* (yksittäisen elementin tunnistaminen, useamman vertailu tai yhteenvedo kaikista). Malli on esitetty kuvassa 14.

### Tutkimistoiminnot

#### Datan tutkimistoiminnot

(Datamuutokset/visuaaliset muutokset)

- Suodatus (filter)
- Tarkastelu (inspect)
- Kysely (query)
- Palauttaminen (restore)

#### Visuaaliset tutkimistoiminnot

(Visuaaliset muutokset)

- Pyyhkäisy (brush)
- Metaforan muutos (change-metaphore)
- Alueen muutos (change-range)
  - Zoomaus (zoom)
  - Panorointi (pan)
- Yhdistäminen (merge)
- Lajittelu (sort)
- Erottaminen (split)

*Kuva 13: Gotzin ja Zhoun [2008] taksonomia*

### Näkemystoiminnot

(Muistiinpanojen muutokset)

#### Visuaaliset näkemystoiminnot

- Huomautuksilla varustaminen (annotate)
- Kirjanmerkkien teko (bookmark)

#### Tietämyksen näkemystoiminnot

- Luominen (create)
- Muokkaus (modify)
- Poistaminen (remove)

### Metatoiminnot

(Historiamuutokset)

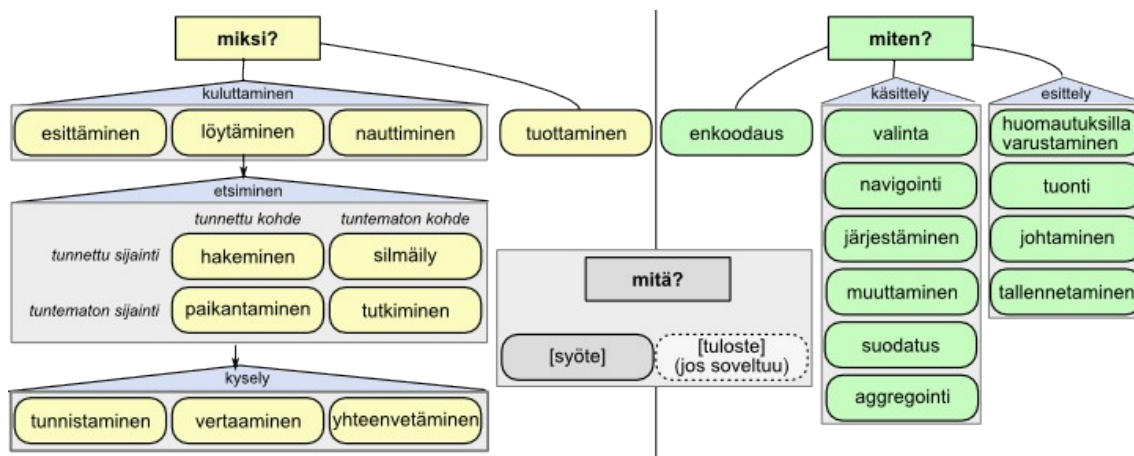
- Poisto (delete)
- Muokkaus (edit)
- Uudelleen tekeminen (redo)
- Palaaminen (revisit)
- Kumoaminen (undo)

Kategoria	Aie
Valinta	Merkitse jotakin kiinnostavaksi
Tutkiminen	Näytä jotakin muuta
Uudelleen järjestäminen	Näytä erilainen asettelu
Muuntaminen (encode)	Näytä erilainen esitystapa
Abstrahointi/tarkennus	Näytä enemmän tai vähemmän yksityiskohtia
Suodatus	Näytä jotakin ehdollisesti
Yhdistäminen	Näytä yhteenkuuluvia asioita

*Taulukko 1: Yin ja muiden [2007] vuorovaikutustekniikoiden kategoriat*

On toki ymmärrettävää, että käyttäjän tarkoitus voidaan käsittää eri tavoin, koska se voidaan hahmottaa hyvin eri tasoilla. Aiemmin esitettyjä prosessimalleja ajatellen käyttäjällä on korkeimman tason tavoitteena saavuttaa tietämystä jostakin aiheesta. Tähän tavoitteluun liittyy esimerkiksi tarkoitus vahvistaa jokin hypoteesi. Analyytisessä prosessissa tavoitteet purkautuvat näin pienempiin ja lopulta päästään tasolle, jossa tarkoi-

tus konkretisoituu käyttäjän fyysiseksi toiminnaksi: hiiren liikuttamiseksi tai painikkeen painamiseksi.



Kuva 14: Brehmerin ja Munznerin [2013] typologia

Esimerkki toiminnoittain jaetusta visuaalisen analytiikan tehtäväluokittelusta on Heerin ja Shneidermanin [2012] vuorovaikutteisen dynamiikan taksonomia, joka on esitetty taulukossa 2. Siinä toiminnot on jaettu kolmeen korkeamman tason luokkaan: datan ja näkymän määrittäminen, näkymän muuntaminen sekä prosessi ja alkuperä. Von Landesberger ja muut [2014] puolestaan näkevät sopiviksi korkean tason kategorioiksi visuaalisen analytiikan osa-alueet visualisointi, päättely ja datan prosessointi. Jokaiseen osa-alueeseen liittyy skeema, parametrit ja data, ja vuorovaikutus liittyy niiden muutoksiin. Von Landesberger ja muut eivät oikeastaan esitä taksonomiaansa sen yksityiskohtaisuuden, vaan antavat vain esimerkkejä eri muutoksista. Nämä on koottu taulukkoon 3.

Riippumatta luokittelun lähtökohdista, luokittelussa päädytään pitkälti samoihin matalamman tason tekniikoihin, tehtäviin tai toimintoihin, mitä termiä nyt haluaakin käyttää. *Valinta* (select) löytyy kaikista, se tarkoittaa kohteen merkintää tai korostamista. *Suodatus* (filter) on myös jokaisessa, ja von Landesberger ja muut tekevät lisäksi sellaisen jaottelun, että suodattaa voi joko prosessointivaiheessa tai visualisoinnissa. Muissa tekniikoissa eri malleissa käytetyt termit poikkeavat jonkin verran toisistaan, mutta asiasisältö on useissa sama. Visualisoinnissa täytyy voida *navigoida* (navigate [Brehmer and Munzner, 2013; Heer and Shneiderman, 2012], change-range: zoom/pan [Gotz and Zhou, 2008], explore, zoomauksen osalta abstract/elaborate [Yi et al., 2007]), joka pitää sisällään vierityksen ja zoomauksen kaltaiset toiminnot. Elementtejä on tarve *järjestää* (sort [Gotz and Zhou; Heer and Shneiderman], arrange [Brehmer and Munzner], reconfigure [Yi et al.]). Järjestäminen tarkoittaa visualisoinnin elementtien avaruudellisen sijainnin muuttamista. Kyseeseen tulee vaikkapa taulukkolaskentaohjelman sarakkeen järjestäminen tai graafin asettelu uudestaan. Tutkiminen vaatii myös *abstraktiotason vaihtelua* (merge/split [Gotz and Zhou], aggregate [Brehmer and Munzner], derive [Brehmer

and Munzner; Heer and Shneiderman], abstract/elaborate [Yi et al.]). Se voi olla visuaalisten elementtien yhdistämistä tai datan yhdistämistä laskennallisesti, mutta toki myös jälkimmäinen heijastuu visualisointiin.

Datan ja näkymien määrittely	<b>Visualisoi</b> data valitsemalla visuaaliset esitystavat
	<b>Suodata</b> pois dataa oleelliseen keskittymiseksi
	<b>Lajittele</b> kohteet hahmojen paljastamiseksi
	<b>Johda</b> arvoja tai malleja datasta
Näkymien käsittely	<b>Valitse</b> kohteita korostettavaksi, suodatettavaksi tai käsiteltäväksi
	<b>Navigoi</b> korkean tason hahmojen ja matalan tason yksityiskohtien tutkimiseksi
	<b>Koordinoi</b> näkymät yhdistetyksi, moniulotteiseksi tutkimiseksi
	<b>Järjestä</b> useat ikkunat ja työtilat
Menetelmä ja alkuperä	<b>Tallenna</b> analyysihistoriat palaamista, arvioimista ja jakamista varten
	<b>Merkitse</b> hahmot löytöjen dokumentoinniksi
	<b>Ja</b> näkymät ja merkinnät yhteistyön mahdollistamiseksi
	<b>Opasta</b> käyttäjät analyysitehtävien tai -tarinoiden läpi

Taulukko 2: Heerin ja Shneidermanin [2012] vuorovaikutteisen dynamiikan taksonomia

	<b>Skeeman muutokset</b>	<b>Parametrien muutokset</b>	<b>Datamuunnos: editointi</b>	<b>Datamuunnos: muu</b>
<b>Visualisointi</b>	mm. visualisointityypin ja ”mäppäyksen” muutokset	Näkymän muutokset kuten zoomaus, vieritys ja värien muuttaminen	Datan editointi	<b>Valinta:</b> sisältää suodatuksen ja ”details-on-demand”
<b>Päättely</b>	Analyysitavan muutos	Historiatoiminnot kuten tee uudelleen ja kumo	Merkintöjen (annotaatioiden) tekeminen	<b>Analyyttisen prosessin jäljittäminen</b>
<b>Datan prosessointi</b>	Algoritmityyppin muutos	Algoritmin parametrien muutos	Uuden datan luominen dataa prosessoimalla	<b>Valinta:</b> datan suodatus

Taulukko 3: Von Landesbergerin ja muiden [2014] vuorovaikutustaksonomia

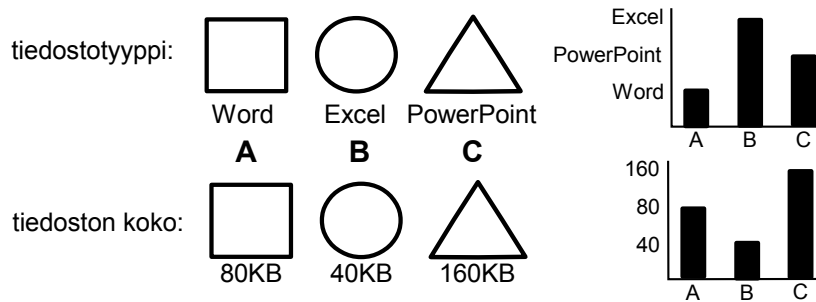
Jotta visualisointi olisi mahdollista, täytyy data jollakin tavalla koodata visuaaliseksi elementeiksi. Tärkeä vuorovaikutustekniikka on myös näiden *koodausten muuttaminen* (visualize [Heer and Shneiderman], encode [Yi et al.], encode, change [Brehmer and Munzner], change-metaphore [Gotz and Zhou]), oli sitten kyse yksinkertaisesta väriskeeman vaihdoksesta tai koko visualisoinnin tyyppin muutoksesta.

Useista luokituksista löytyy myös erilaisia *metadatan tuottamista* koskevia toimintoja (merkintöjen ja muistiinpanojen tekeminen) sekä *historiatoimintoja* (toista/kumoa jne.). Yksittäisiä mainintoja saavat datan tuonti (query [Gotz and Zhou], import [Heer and Shneiderman]), useiden näkymien hallinta (organize, coordinate [Heer and Shneiderman], inspect eli details-on-demand [Gotz and Zhou]), muille ihmisille jakaminen ja opastus [Heer and Shneiderman], prosessointialgoritmin tai sen parametrien muutos [von Landesberger] sekä yhdistäminen (connect [Yi et al.]). Viimeinen tarkoittaa toisiinsa liittyvien elementtien korostamista, esimerkiksi graafissa voisi korostaa kaikki sellaiset solmut, joista on suora yhteys valittuun solmuun.

Vuorovaikutus on siis prosessi, joka alkaa käyttäjän toiminnasta ja päättyy iteratiivisesti tietämykseen. Prosessin iteraatioissa käyttäjä suorittaa tehtäviä vuorovaikutustekniikoiden avulla. Sekä prosessilla että yksittäisellä tehtävällä on alku ja loppu. Sachan ja muiden [2014] mallissa yksittäisen tehtävän voi ajatella päättyväksi löytämiseen, joka tuottaa uusia aikeita (mahdollisesti vahvistus- ja tietämyksen tuottamissilmukoiden kautta) ja siten uusia tehtäviä. Yleisiä tehtäviä ovat valinta, suodatus, navigointi, järjestäminen, abstraktiotason vaihtaminen, visuaalisten koodausten muuttaminen, metadatan tuottaminen sekä historiatoiminnot. Kuten Yi ja muut [2007] huomauttavat, eri tekniikat liittyvät toisiinsa: erityisesti valinta on muiden tekniikoiden apuna.

### 3.3 Esitystavoista

Datan mahdolliset visuaaliset esitystavat riippuvat datan tyyppistä. Yksinkertaisen esimerkin tästä antaa Zhang [1996]: tiedoston tyyppiä voi esittää muodolla, mutta jos tiedoston kokoa yritetään esittää vastaavalla koodauksella, esitystapa jää vajavaiseksi. Muoto on visuaalinen elementti, joka kertoo ainoastaan, että asiat ovat erityyppisiä, mikä on ainut mielekäs tapa käsittää tiedostotyyppit. Tiedostojen ko'oista muoto paljastaa ainoastaan, että ne ovat eri kokoiset, vaikka kokoja voisi verrata esimerkiksi suhteellisesti: tiedosto B on kaksi kertaa niin suuri kuin tiedosto A. Koolle mielekäs esitystapa onkin esimerkiksi pylväsdiagrammi, joka taas olisi harhaanjohtava tapa esittää tiedostotyyppiä: kuvan 15 esimerkki antaa vaikutelman, että PowerPoint-tiedosto olisi kaksi kertaa Word-tiedosto, mikä ei ole millään tasolla mielekäs tulkinta. Zhangin viesti on, että visuaalisen esitystavan dimensioiden on vastattava datan dimensioita. Pylväsdiagrammi sisältää enemmän tietoa kuin datassa tiedostotyyppin suhteen on, kun taas muodot sisältävät vähemmän tietoa kuin datassa koon puolesta on.



Kuva 15: Zhangin [1996] esimerkki

Tyypillinen luokittelu datalle on tilastotieteessä käytetyt Stevensin [1946] neljä asteikkoa: luokittelu-, järjestys-, välimatka- ja suhdeasteikot. Edellisessä esimerkissä on siis kyse luokittelu- ja suhdeasteikkollisen datan eroista. Visualisointiin tämä luokittelu ei kuitenkaan yksin riitä. Ajatellaan vaikkapa tietokantaa ihmisistä, joka sisältää tiedot sukupuoli (luokitteluasteikkollinen), syntymäaika (välimatka-asteikkollinen), ikä (suhdeasteikkollinen) ja koulutusaste (järjestysasteikkollinen). Toki yksittäisiin tietoihin liittyy aina asteikkotyyppi, mutta näiden asteikoiden kuvausvoima lakkaa, kun pitäisi kuvata miten tiedot liittyvät toisiinsa.

Shneiderman [1996] jakaa datan seuraaviin kategorioihin: yksi-, kaksi-, kolme- ja moniulotteinen data, ajallinen data, puu- eli hierarkkinen data ja verkot. Ware [2012, 25-29] puolestaan hahmottelee datan tyypeiksi oliot, suhteet, ulottuvuudet, dataan liittyvä epävarmuus, tietyt operaatiot kuten keskiarvon laskeminen tai olioiden poistaminen sekä metadatan. Jako ei ole yksiselitteinen, olio voi olla kala siinä missä kalaparvikin. Metadata taas on käytännössä tyypiltään joitain muista tyypeistä, kuten olioita tai suhteita.

Kalsiumsignaalit itsessään ovat Waren olioita ja Sheidermanin ajallista dataa. Poikkeamat ovatkin hankalampi tapaus: toisaalta nekin ovat ajallista dataa, onhan poikkeamalla signaalissa sekä alku että loppu, toisaalta taas voi ajatella signaaliin liittyvän useita useammantyyppisiä poikkeamia, jolloin kyse on moniulotteisesta datasta. Koska erityyppiselle datalle on kehitetty runsaasti erilaisia visualisointimenetelmiä, keskitytään tässä ainoastaan ajallisen ja moniulotteisen datan mahdollisiin esitystapoihin.

### 3.3.1 Ajallinen data

Ajallista dataa on monenlaista, ja toimivat visualisointitavat riippuvat siitä, millaista data on. Aignerin ja muiden [2007] mukaan eroavuuksia on niin ajallisessa ulottuvuudessa kuin muussakin datassa. Ajassa voi olla kyse hetkellisistä tapahtumista tai aikaväleistä. Ensimmäisessä tapauksessa tapahtumista voi sanoa vain, että toinen tapahtui ennen toista tai molemmat yhtä aikaa. Aikaväleillä taas on monimutkaisempia suhteita, koska ne voivat limittyä. Lisäksi aikaa voi hahmottaa lineaarisesti tai syklistä, esimerkiksi historia nähdään tyypillisesti lineaarisena, kun taas vuodenkierto on syklinen.

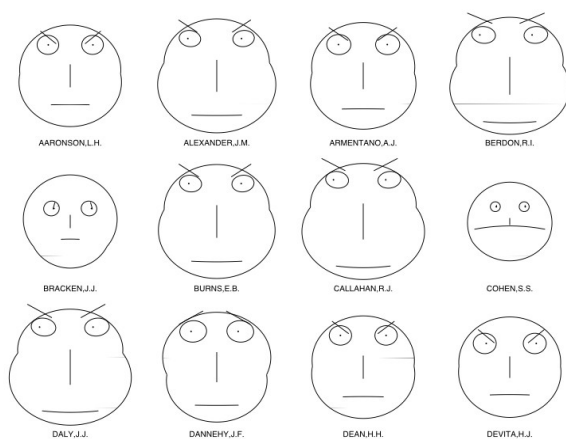
Muun datan Aigner ja muut jakavat abstraktiin ja avaruudelliseen, mikä on hyvin luonnollista: visualisointitavan kannalta on oleellisesta, onko data sidottu paikkaan. Esimerkiksi lentoliikennettä tutkiessa voi olla tärkeää nähdä, missä koneet sijaitsevat. Abstraktille datalle voi keksiä visualisointeja vapaammin kuin paikkasidonmaiselle datalle. Myös datan ulottuvuudet vaikuttavat: data voi sisältää yhden tai useita muuttujia ajan suhteen.

Signaalit ovat tyypillisesti dataa, jossa on vain yksi muuttuja ajan suhteen. Se pätee myös kalsiumsignaaliin, joka kuvaa kalsiumionipitoisuuden muutosta ajassa. Sen vuoksi signaaleita visualisoidaan yleensä viivadiagrammilla. Kalsiumsignaalien tapauksessa sen käyttöä puoltaa myös se, että asiantuntijat ovat tottuneet tunnistamaan poikkeamat nimenomaan käyrän muotoa katsomalla. Muunlainen visualisointi heikentäisi käyttäjien kykyä arvioida ohjelman antamia tuloksia.

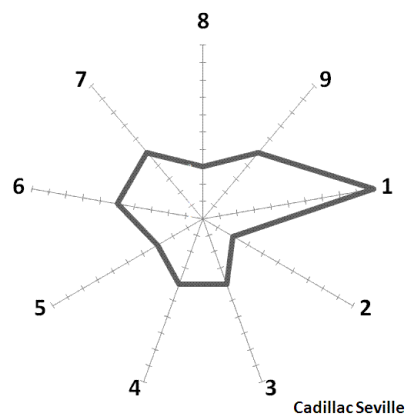
### 3.3.2 Moniulotteinen data

Moniulotteisen datan visualisointikeinot voi luokitella viiteen kategoriaan: ikoneihin, pikselipohjaisiin, geometrisiin, hierarkkisiin ja graafiperusteisiin tekniikoihin [Keim, 2000]. Hierarkkiset menetelmät ja graafit eivät kuitenkaan ole kaikelle moniulotteiselle datalle soveltuvia tekniikoita, vaan ne vaativat datalta tiettyä rakennetta, puuta tai verkkoa. Signaalidata ei ole kumpaakaan, joten näitä tekniikoita ei tässä käsitellä.

Kun käytetään ikoneita (glyph, icon), data esitetään joukkona pieniä, toisistaan erillisiä visuaalisia objekteja, joista kukin kuvaa yhden data-alkion attribuutit. Attribuuttien arvoja voidaan kuvata muodolla, värillä, tekstuurilla, koolla, suuntauksella tai kaarevuudella. [Borgo et al., 2013] Tunnettuja ikoneita ovat esimerkiksi Chernoffin kasvot (Kuva 16) ja tähti-ikonit (Kuva 17). Chernoffin kasvoissa jokainen piirre, kuten nenä tai suu, vastaa yhtä attribuuttia, ja piirteen koko tai muoto kertoo attribuutin arvon. Tähti-ikonissa kullekin attribuutille on oma, kuvion keskeltä alkava akselinsa. Data-alkion attribuuttien arvot eri akseleilla yhdistetään viivalla.

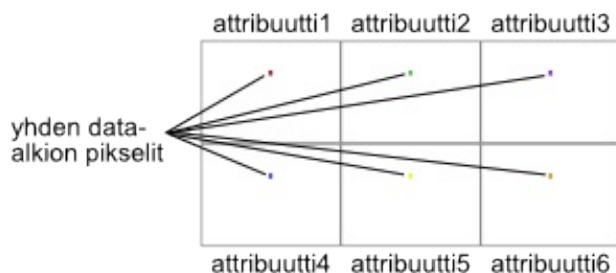


Kuva 16: Chernoffin kasvot



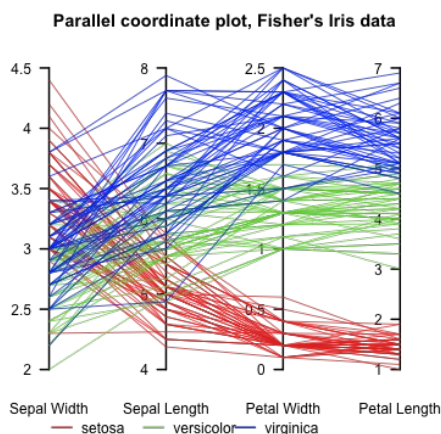
Kuva 17: Tähti-ikoni

Pikselipohjaisissa tekniikoissa näkymä jaetaan useisiin ikkunoihin, yksi kutakin attribuuttia kohti. Kutakin data-alkiota vastaa tietty pikseli jokaisessa ikkunassa ja kutakin attribuutin arvoa tietty väri (Kuva 18). Tekniikat vaihtelevat värimääritysten, pikselien järjestyksen ja ikkunan muodon mukaan [Keim, 2000].

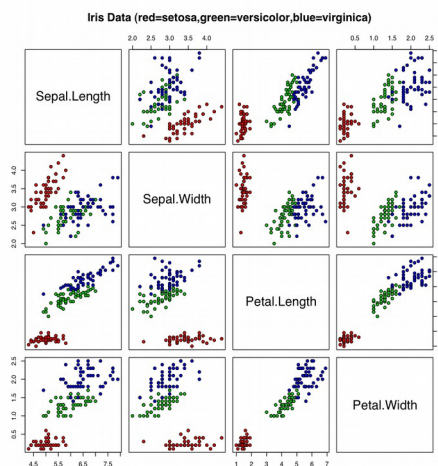


Kuva 18: Pikselipohjaisten tekniikoiden perusajatus

Geometrinen tekniikoiden ideana on erilaiset projektiot data-avaruudesta, kuten rinnakkaiskoordinaatit ja hajontakuviomatriisit. Rinnakkaiskoordinaateissa (Kuva 19) kullekin attribuutille on oma pystysuora akselinsa ja nämä akselit piirretään rinnakkain. Kutakin data-alkiota vastaa murtoviiva, joka yhdistää alkion arvot eri akseleilla. Hajontakuviomatriisissa (Kuva 20) piirretään kaikista kahden attribuutin yhdistelmästä normaalit hajontakuviot ja järjestetään ne matriisiksi.



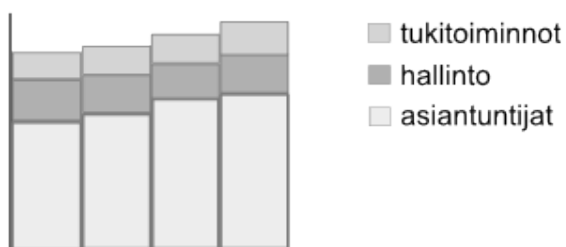
Kuva 19: Rinnakkaiskoordinaatit



Kuva 20: Hajontakuviomatriisi

Edellä esitettyjä tapoja voisi soveltaa myös kalsiumsignaalien poikkeamiin, mutta toisaalta kalsiumsignaalien tapauksessa visualisointia helpottaa se, että kaikki attribuutit ovat samantyyppisiä. Poikkeamista voidaan toki laskea erilaisia suureita, kuten esimerkiksi poikkeamien kappalemäärä tai kuinka suuren osuuden ajasta signaali on poikkeava, mutta jokaiselle signaalille tehdään samat laskutoimitukset. Ei olisi mitään mieltä vertailla yhden signaalien poikkeamamäärää toisen signaalien ajalliseen poikkeavuuteen.

Attribuuttien yhteneväisyys mahdollistaa tekniikat, joita ei voi käyttää, jos attribuuttien datatyypit vaihtelevat. Esimerkki sellaisesta tekniikasta on pinottu pylväskaavio (Kuva 21). Siinä missä yksinkertainen pylväskaavio kuvaa yhden muuttujan lukuarvoja, vaikkapa henkilöstön määrää yrityksessä, pinotussa pylväskaaviossa voidaan esittää useampia samankaltaisia muuttujia kerralla laittamalla arvoja vastaavat pylväät päällekkäin, jolloin voidaan esimerkiksi esittää myös henkilöstön määrä ryhmittäin.



Kuva 21: Esimerkki pinotusta pylväskaaviosta

### 3.3.3 Useat näkymät

Visualisoinnissa voidaan käyttää useita näkymiä, kun halutaan tarjota erilaisia näkökulmia tai välttää näkymän muuttuminen sekavaksi elementtien paljouden seurauksena. Tyypillisiä käyttötapoja useille näkymille ovat esimerkiksi yleiskuva ja yksityiskohdat tai vertailu. Yleisesti tunnettu esimerkki useista näkymistä on Google Maps -karttapalvelun Street View -toiminto, jolla voi tutkia Googlen kuvaamia katunäkymiä: yhdessä näkymässä on katukuva ja toisessa kartta.

Robertsin [2007] mukaan useilla näkymillä tarkoitetaan yleensä peräkkäisiin ikkunoihin sijoitettuja erilaisia esitystapoja, joihin kohdistuvat toiminnot ovat koordinoituja. Tällöin voidaan puhua myös useista koordinoituista näkymistä. Baldonado ja muut [2000] määrittelevät yksittäisen näkymän datajoukkona ja määrittäen sen visuaaliselle esitystavalle. Näkymät ovat erillisiä, jos ne mahdollistavat eri piirteiden tutkimisen. Ne voivat olla erilaisia joko datan tai sen esitystavan kannalta.

Visualisointien sijoittaminen allekkain tai päällekkäin, kuten useille näkymille on tyypillistä, ei kuitenkaan ole ainut vaihtoehto. Eri näkymät voi erottaa myös ajallisesti eli esittää yhden näkymän kerrallaan. Tällöinkin näkymien toiminta on yhdistettävissä. Esimerkiksi julkista liikennettä visualisoivassa Touching Transport -sovelluksessa [Nagel et al., 2014] eri näkymissä voi valita bussipysäkkejä ja valinnat pysyvät näkymää vaihtaessa. Näkymien vaihtuminen on myös animoitu käyttäjän hahmotuksen tukemiseksi ja kontekstin säilyttämiseksi.

Toisaalta visualisointeja voi asettaa eri tavoin myös päällekkäin tai sisäkkäin, vaikkei useilla näkymillä tyypillisesti tarkoiteta sellaista. Javed ja Elmqvist [2012] ehdottavatkin uutta käsitettä yhdistelmävisualisointinäkymät (composite visualization views). Tämä termi kattaa muitakin tapoja näkymien yhdistämiselle kuin peräkkäisten,



toisistaan erillisten näkymien toiminnallinen koordinointi. Nämä tavat Javedin ja Elmqvistin luokittelulla ovat peräkkäisyys (juxtaposition), päällekkäisyys (superimposition), ylikuormitus (overload), sisäkkäisyys (nesting) ja yhdistäminen (integration).

Peräkkäiset näkymät tarkoittavat visualisointien asettamista rinnakkain tai allekkain. Päällekkäisissä näkymissä visualisoinnit ovat kerroksittain toistensa päällä, ainakin osittain läpinäkyvinä, esimerkiksi erilaiset merkinnät kartalla. Olennaista on myös, että näkymät vastaavat toisiaan avaruudellisesti yksi yhteen, kuten karttamerkintöjen ja niitä vastaavien kohteiden sijainnit. Ylikuormituksessa isäntänäkymän sisään tai päälle piirretään asiakasnäkymä, mutta toisin kuin päällekkäisissä näkymissä, visualisoinneilla ei ole yksi yhteen vastaavuutta. Sisäkkäisissä näkymissä asiakasnäkymät ovat isäntänäkymän visuaalisten merkintöjen sisällä: esimerkiksi hajontakuvion pisteet ovatkin pylväsdiagrammeja. Yhdistetyissä näkymissä erilliset näkymät on yhdistetty visuaalisilla linkeillä, jotka yhdistävät eri näkymien toisiaan vastaavat visuaaliset merkit.

Jotta useista näkymistä olisi hyötyä, täytyy niitä koordinoida. Se tarkoittaa, että yhteen näkymään kohdistettu toiminto heijastuu myös toiseen näkymään, esimerkiksi jos valitaan kohde yhdestä näkymästä, korostuu sama kohde toisessakin näkymässä. Näkymiä voi koordinoida monin eri tavoin, eikä muissa näkymissä tapahtuvien muutosten tarvitse olla samoja, kuin toiminnon kohteena olleessa näkymässä. Muut näkymät voivat muuttua kokonaankin: yleiskuvasta voi valita jonkin kohteen, jolloin kohteen yksityiskohtia esittävä näkymä päivittyy valitun kohteen mukaiseksi.

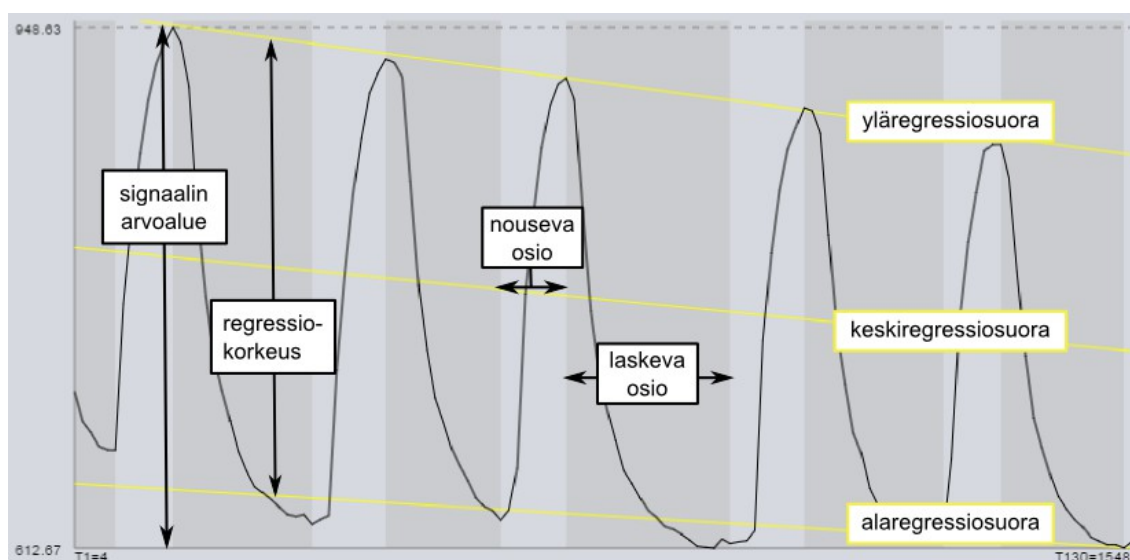
Googlen Street View käy esimerkiksi myös näkymien koordinoinnista: Kartalla näkyy merkki ja nuoli esittämässä katunäkymän kohtaa ja suuntaa. Katunäkymässä voi liikkua joko valitsemalla uuden kohdan näkymästä itsestään tai siirtämällä merkkiä karttanäkymässä.

Kalsiumsignaalien suhteen yhtenä tavoitteena oli vertailutoiminto lääkevasteiden tutkimista varten. Vertailu on yksi niistä toiminnoista, joissa useita näkymiä voi hyödyntää, ja sitä onkin hyödynnetty tässä esiteltävässä ratkaisussa. Lisäksi ratkaisussa käytetään hyvin tyypillistä tapaa erottaa yleiskuva ja yksityiskohdat omiin näkymiinsä. Näkymät esitellään yksityiskohtaisesti seuraavassa luvussa.

## 4. Ratkaisun kuvaus

Koska poikkeamien luokittelu on vaikeaa puhtaasti laskennallisilla menetelmin, valittiin tässä ratkaisuksi vuorovaikutteinen visualisointi. Siinä laskennasta on tehty parametri-soitu eli käyttäjä voi säätää tiettyjä arvoja, jotka vaikuttavat laskennan lopputulokseen. Visualisointi auttaa käyttäjää ymmärtämään, miten parametrien eri arvot vaikuttavat lopputulokseen. Parametrien muuttamisen vaikutus näkyy välittömästi visualisoinnissa.

Poikkeamien laskenta tapahtuu siten, että ensin jaetaan signaali nouseviin ja laskeviin osiin. Käyttäjän on mahdollista säätää paljonko osioiden laskennassa huomioidaan kohinaa. Sen jälkeen signaalille lasketaan osioiden huippujen ja pohjien perusteella kolme regressiosuoraa: ylä-, keski- ja alasuorat. Ylä- ja alasuorien erotusta kutsutaan regressiokorkeudeksi. Termit on esitetty myös kuvassa 22. Poikkeamat lasketaan enimmäkseen regressiokorkeuden ja osioiden korkeuden perusteella huomioiden parametrien arvot. Kaikki parametrit ovat prosentuaalisia arvoja. Kohinan parametri on prosenttiosuus signaalin arvoalueesta eli suurimman ja pienimmän arvon erotuksesta.



Kuva 22: Signaaliin liittyviä termejä

### 4.1 Regressiosuorien laskenta

Regressioanalyysi on tilastotieteellinen keino tutkia selitettävän muuttujan riippuvuutta selittävistä muuttujista. Tässä selittävä muuttuja on aika ja selitettävä muuttuja on signaalin arvo. Sekä signaalin huippujen että pohjien oletetaan olevan lineaarisesti riippuvia ajasta. Huipuille ja pohjille lasketaan erikseen regressiosuorat. Lisäksi näiden suorien perusteella lasketaan keskimäinen regressiosuora. Koska poikkeamia sisältävissä signaaleissa voi olla huomattavastikin yleisestä trendistä poikkeavia arvoja, täytyy näitä poikkeavuuksia pyrkiä karsimaan suorien laskennassa, jotta ne eivät pääse vääristämään tuloksia.

Ylä- ja alaregressiosuorat lasketaan kaksivaiheisesti: Ensimmäisessä vaiheessa valitaan tietyt huiput tai pohjat ja lasketaan niiden perusteella regressiosuorat. Toisessa vaiheessa valitaan huiput tai pohjat uudelleen ensimmäisessä vaiheessa laskettujen regressiosuorien perusteella, ja lasketaan sitten suorat niiden perusteella.

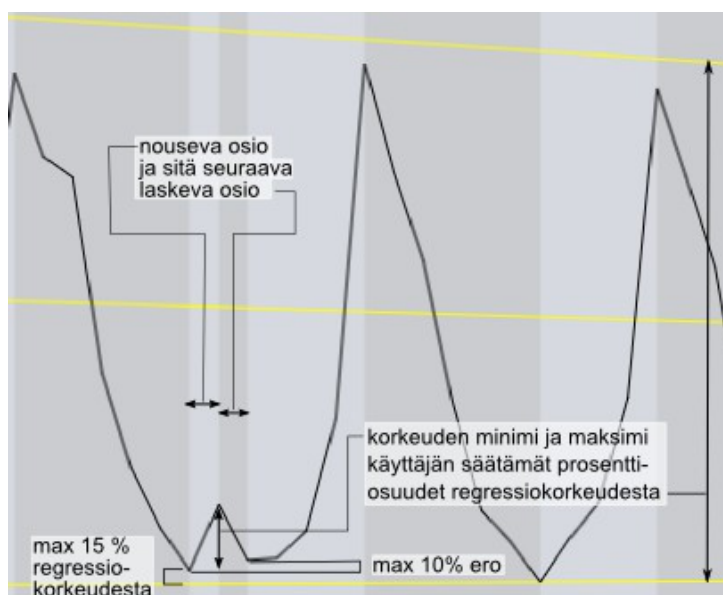
Yläregressiosuoran ensimmäisen vaiheen laskentaan otetaan mukaan sellaiset osiot, joiden korkeus on yli 60% signaalin osien keskimääräisestä korkeudesta tai joiden minimiarvo on signaalin maksimia pienempi korkeintaan 40% signaalin arvoalueesta. Toisessa vaiheessa huomioidaan ne huiput, joiden arvo on suurempi kuin ensimmäisessä vaiheessa lasketun regressiokäyrän arvo huipun kohdalla.

Alaregressiosuoran ensimmäisessä vaiheessa valitaan sellaiset osiot, joiden korkeus on suurempi kuin 75% signaalin matalimmasta osiosta tai joiden minimiarvo poikkeaa signaalin minimistä alle 20% signaalin korkeudesta. Toisessa vaiheessa otetaan mukaan ne osiot, joiden minimiarvo on pienempi kuin ensimmäisen vaiheen regressiosuoran arvo minimin kohdalla.

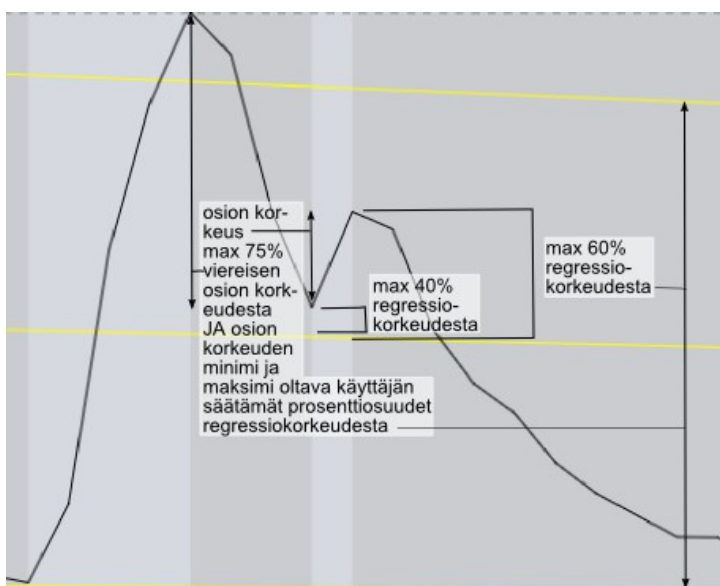
Keskiregressiosuora lasketaan ylä- ja alaregressiosuorien molempien vaiheiden jälkeen. Se lasketaan näiden suorien väliin yksinkertaisesti laskemalla kulmakertoimien ja vakiotermin keskiarvot.

#### 4.2 Poikkeamien laskenta

Matalaksi piikiksi lasketaan sellainen nouseva osio (ja sitä seuraava laskeva osio), jonka minimiarvo on alaregressiokäyrän arvoa suurempi korkeintaan 15% regressiokorkeudesta. Lisäksi laskevan osion minimin arvo saa poiketa nousevan osion minimistä korkeintaan 10%. Matalaan piikkiin liittyy kaksi parametria: minimi- ja maksimiosuus regressiokorkeudesta, joiden väliin osion korkeuden on kuuluttava. Matalan piikin laskeminen on esitetty myös kuvassa 23.

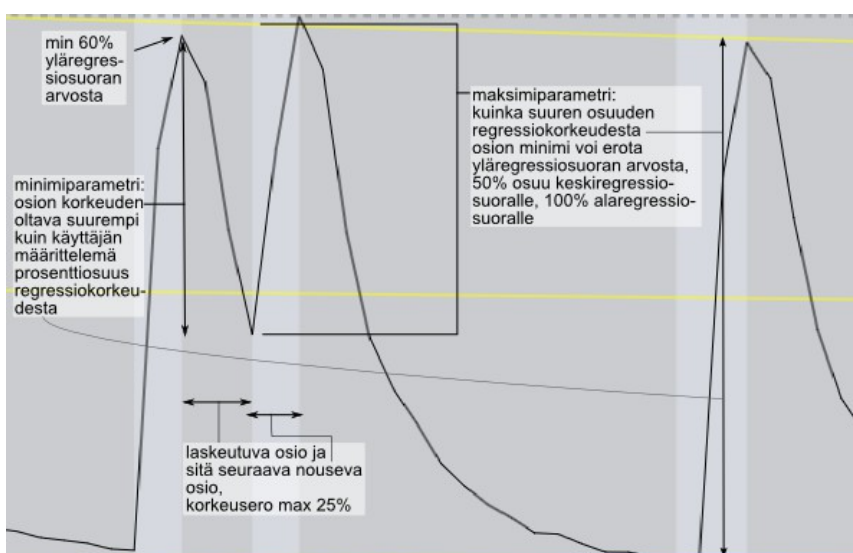


Kuva 23: Matalan piikin laskeminen



Kuva 24: Keskipiikin laskeminen

Keskipiikin maksimi saa poiketa keskiregressiokäyrän arvosta korkeintaan 60% ja minimi 40%. Osion korkeuden täytyy olla pienempi kuin 75% viereisen osion korkeudesta. Keskipiikin parametrit ovat samat kuin matalalla piikillä: minimi- ja maksimiosuus regressiokorkeudesta. Kuvassa 24 on esitetty keskipiikin tunnistaminen, kun keskipiikki alkaa nousevalla osiolla. Tällöin viereinen osio, johon korkeutta verrataan, tarkoittaa edellistä laskevaa osiota. Kun keskipiikki alkaa laskevalla osiolla (eli kyseessä on peilikuva kuvan 24 tapauksesta), viereinen osio tarkoittaa seuraavaa nousevaa osiota.



Kuva 25: Monihuippuisen piikin laskeminen

Monihuippuinen piikki on laskeutuva osio (ja sitä seuraava nouseva osio), jonka maksimi on vähintään 60% yläregressiokäyrän arvosta ja korkeus  $\pm 25\%$  seuraavan nousevan osion korkeudesta. Tämän lisäksi piikkiin liittyy kaksi parametria. Osion minimi

saa olla yläregressiokäyrän arvoa pienempi korkeintaan annetun prosenttiosuuden verran regressiokorkeudesta. Osion korkeuden on oltava suurempi kuin annettu osuus regressiokorkeudesta. Monihuippuisen piikin laskemiseen liittyvät seikat on esitetty myös kuvassa 25.

Alapiikki on kaikista yksinkertaisin laskea. Se on laskeutuva osio (ja sitä seuraava nouseva osio), jonka minimiarvo on alaregressiokäyrän alapuolella käyttäjän määrittämän osuuden verran regressiokorkeudesta.

Tasopoikkeaman laskenta perustuu laskevaan tai nousevaan osioon kuuluvien mittausarvojen muutokseen. Mittausarvojen muutoksia tarkastellaan kolmen peräkkäisen arvon ryhmissä. Jos tällaisen kolmikön muutosnopeus putoaa välillä käyttäjän säätämän prosenttiosuuden verran aiemmasta ja kuitenkin muutosnopeus loppuosion aikana palaa taas ennalleen, katsotaan kyseessä olevan tasopoikkeama.

Epäsäännöllisten vaiheiden tunnistamista varten lasketaan signaalin huippujen välien mediaani. Mediaanin laskennassa jätetään huomiotta matalat ja keskihiikit. Epäsäännöllisiin vaiheisiin liittyy vain yksi parametri: osuus mediaanista, jota suuremmat poikkeamat katsotaan epäsäännöllisiksi.

### **4.3 Tekniikat**

Toteutuksessa haluttiin välttää mahdolliset ongelmat ohjelman asennuksessa, esimerkiksi käyttäjältä puuttuvat asennusoikeudet päätteelleen. Lisäksi haluttiin, että ohjelma olisi helposti siirrettävissä eri alustoille. Verkkosivu oli siten luonnollinen valinta. Se myös mahdollistaa toimivuuden sellaisenaan eri käyttöjärjestelmissä ja laitteissa. Koska ei ollut tarvetta tukea vanhoja selainversioita, teknologioiksi valikoitui HTML5 ja JavaScript. Koska käsiteltävät signaalit ovat lyhyitä eli dataa ei ole valtavaa määrä kerrallaan ja poikkeamien laskenta ei ole erityisen raskasta, ei ollut tarvetta toteuttaa erillistä palvelinpuolta, vaan kaikki datan käsittely ja laskenta tehdään selaimessa.

#### **4.3.1 HTML**

HTML (Hypertext Markup Language) on merkintäkieli, jolla kuvataan WWW-sivujen rakenne. HTML-standardista vastaa W3C (World Wide Web Consortium), joka on taannoin julkaissut standardista uuden version 5 [HTML5]. Se oli kehitteillä vuosia, ja useisiin selaimiin on toteutettu sen ominaisuuksia standardin kehitysprosessin edetessä. Tämä on mahdollistanut HTML5-pohjaisten sivustojen kehittämisen jo kauan ennen standardin valmistumista. Tähän mahdollisuuteen onkin tartuttu, sillä edellinen HTML-versio, 4.01, julkaistiin 1999, eikä se ole aikoihin vastannut kehittäjien ja käyttäjien tarpeita.

Versio 5 poistaa HTML:stä turhiksi käyneitä elementtejä ja lisää uusia, esimerkiksi video- ja audioelementit multimediasisältöjä varten ja canvas- ja svg-elementit grafiikkaa varten. Lisäksi HTML5 sisältää uusia ohjelmointirajapintoja, esimerkiksi mainittu-

jen videon ja audion toiston hallintaan. Uudet ominaisuudet vähentävät tarvetta käyttää tekniikoita, jotka vaativat käyttäjältä selaimiin lisättäviä lisäosia, kuten esimerkiksi Flash, jota on aiemmin käytetty muun muassa videosisältöjen tekemiseen.

Toteutetussa sovelluksessa onkin hyödynnetty HTML5:n uutta canvas-elementtiä, jolle visualisoinnit piirretään. Vaihtoehtoinen elementti olisi svg, jota käytetään vektorigrafiikkaan, kun taas canvas edustaa pikseligrafiikkaa. Canvaksen liittyy context-piirto kontekstiolio, joka sisältää varsinaisen rajapinnan canvakselle piirtämiseen. HTML5-spesifikaatio määrittää erilaisia kontekstiolioita, joista ”2d” pitäisi olla aina käytettävissä. Sen lisäksi on ”webgl”, jota selaimen ei ole pakko tukea. Selainvalmistajille on myös jätetty mahdollisuus määritellä omia kokeellisia kontekstejaan.

Ratkaisussa on käytetty 2d-kontekstia, joka nimensä mukaisesti tarjoaa rajapinnan kaksiulotteiseen piirtoon. Kolmiulotteinen piirto on mahdollista WebGL-kontekstin avulla, mutta se ei ollut tässä tarpeen. 2d-konteksti tarjoaa peruspiirtotoimintoja: viivoja, polkuja, nelikulmioita ja tekstiä, sekä näille erilaisia tyylejä, kuten väri ja viivanpaksuus. Lisäksi 2d-kontekstilla on transformaatiomatriisi, jonka avulla voi skaalata, kääntää ja siirtää canvakselle piirrettäviä primitiivejä.

### 4.3.2 JavaScript

JavaScript on ohjelmointikieli, joka kehitettiin mahdollistamaan dynaamisuuden WWW-sivuissa. JavaScriptillä voi siis päivittää HTML-sivujen elementtejä lataamatta koko sivua uudestaan. Tämä mahdollistaa vaikkapa lomakkeen validoinnin ennen sen lähettämistä palvelimelle. Nykyisin JavaScriptiä käytetään monissa muissakin yhteyksissä, esimerkiksi palvelinpuolella (node.js).

JavaScript kehitettiin alun perin Netscape Navigator -selainta varten 1990-luvulla. Microsoft teki pian Internet Explorer -selaimensa oman toteutuksensa JavaScriptistä nimellä JScript, vaikka nykyisin sekin käyttää nimeä JavaScript. Myöhemmin kielestä tehtiin standardi, jossa se nimettiin kompromissina ECMAScriptiksi tieto- ja viestintätekniologian standardointijärjestö Ecma Internationalin mukaan. ECMAScript-spesifikaatioon perustuu nykyään muitakin kieliä, kuten Adoben ActionScript 3. Nykyinen standardin versio 5.1 on vuodelta 2011 [ECMA-262], mutta kehitystyö jatkuu edelleen.

ECMAScriptin spesifikaatio sallii standardia noudattavien kielten toteuttavan myös lisäominaisuuksia, joita spesifikaatioissa ei määritellä. Tästä seuraa, että JavaScriptin eri toteutukset saattavat poiketa toisistaan. Netscapen perillinen Mozilla versioikin JavaScriptiä erikseen vuoteen 2010 asti, mutta on luopunut käytännöstä. Selainpohjaisten ohjelmistojen kehitystä helpottaa kuitenkin W3C:n erilaiset valmiit ja valmisteilla olevat standardit, joissa määritellään JavaScript-ohjelmointirajapintoja selaimiin.

JavaScript on moniparadigmainen, dynaamisesti tyyplitetty kieli. Se tukee proseduraalista, funktionaalista ja olio-ohjelmointia. JavaScriptin funktionaalisuus näkyy siinä,

että funktioita voi asettaa muuttujien arvoksi, välittää funktion parametrina tai palauttaa toisesta funktiosta.

Toisin kuin useissa muissa yleisissä oliokielissä, JavaScriptin olio-ohjelmointi ei perustu luokkiin vaan prototyyppeihin. Siinä missä luokka on olioille yhteinen määrittely, prototyyppi itsessään on olio, josta luodaan uusia olioita. JavaScriptissä oliot muodostavat ominaisuusketjuja. Kun olion ominaisuuteen viitataan, ensisijaisesti tutkitaan, löytyykö oliolta itseltään sen nimistä ominaisuutta. Mikäli ei löydy, seuraavaksi tutkitaan olion prototyyppi, josta taas jatketaan prototyypin prototyyppiin ja niin edelleen kunnes ominaisuus löytyy tai saavutetaan ketjun pää.

Toteutetussa ohjelmistossa käytetään JavaScriptiä laskentalogiikkaan ja vuorovaikutukseen. Toteutuksessa on lisäksi hyödynnetty JavaScript-kirjastoa jQuery peittämään joitakin selainten eroavaisuuksia sekä jQuery UI:n liukusäätimiä.

#### 4.4 Näkymät

Visualisointi koostuu kolmesta varsinaisesta näkymästä: yhteenveto, datanäkymä ja analyysinäkymä, jossa voi tarkastella yksittäistä signaalia yksityiskohtaisesti. Yhteenvetonäkymä on ensimmäisenä, kuten Shneiderman [1996] tunnetussa tiedon visualisointi-mantrassaan ("Overview first, zoom and filter, then details-on-demand") kehottaa. Yhteenveto- ja analyysinäkymät muodostavat tyypillisen useiden näkymien yhdistelmän: yleiskuva ja yksityiskohdat (overview+detail).

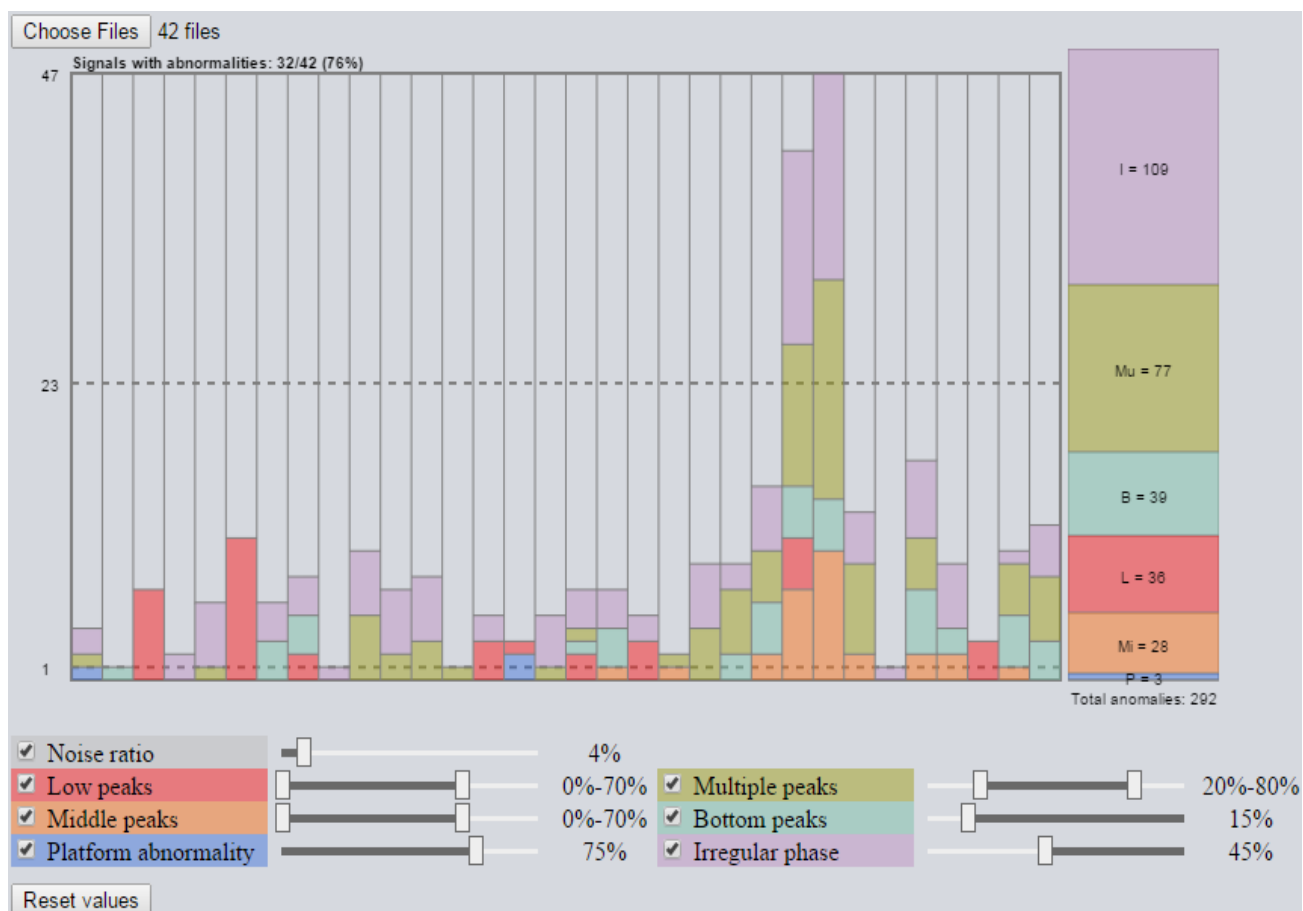
Lisäksi käyttöliittymässä on parametrinäkymä: liukusäätimet kohinan ja kunkin poikkeamatyyppin laskennassa käytettäville parametrialvoille (Kuvan 26 alareuna). Kunkin poikkeamaan liittyy yksi tai kaksi parametria, jotka kuvattiin tarkemmin kohdassa 4.2. Kaksi parametrialvoa liittyy aina saman asian minimiin ja maksimiin, joten ne on yhdistetty samaan kaksipuoleiseen liukusäätimeen. Kutakin poikkeamaa vastaa siten yksi liukusäädin. Säädetyt parametrialvot näytetään myös numeerisina liukusäädinten vieressä.

Parametrien liukusäätimien yhteydessä esitetyillä poikkeaman nimillä on taustavärit. Nämä taustat yhdistävät tietyn värin kuhunkin poikkeamaan, ja samoja värejä käytetään kaikissa näkymissä kuvaamaan poikkeamatyyppiä.

Varsinaisen visualisoinnin ja sen säätimien lisäksi ohjelmaan kuuluu vientinäkymä, josta on mahdollista kopioida senhetkiset tulokset taulukkolaskentaohjelmaan tai muuhun vastaavaan. Näkymässä on listattu kaikki signaalit ja niissä olevat poikkeamien määrät tyypeittäin sarkaimella erotettuna, joten ne kopioituvat siististi omiin soluihinsa taulukkolaskentaohjelmiin. Näkymää ei ole tarkoitettukaan datan tutkimiseen ohjelman itsensä puitteissa, vaan pelkästään tulosten siirtoon, jos on tarvetta jonkinlaiselle jatkokäsittelylle.

#### 4.4.1 Yhteenvetonäkymä

Yhteenvetonäkymä (Kuva 26) tarjoaa yleiskuvan signaaleista ja erityisesti poikkeamista. Yhteenveto koostuu pylväskaaviosta ja yhdestä erillisestä pylväästä. Pylväskaaviossa esitetään kaikki sellaiset signaalit, joissa esiintyy poikkeamia. Jokaisella sellaisella signaalilla on oma pylväänsä, josta näkyy poikkeamien määrä ja niiden tyypit. Pylvään korkeus kuvaa kokonaismäärää ja eriväriset osuudet pylväästä eri poikkeamatyyppien osuuksia.



Kuva 26: Yhteenveto- ja parametrinäkymät

Normaaleiksi luokiteltuja signaaleja, joissa ei siis ole yhtään poikkeamaa, ei yhteenvedossa esitetä. Näkymässä kerrotaan, kuinka monta epänormaalia signaalia on sekä paljonko signaaleja data kaikkiaan sisältää. Lisäksi näkyy, kuinka suuri prosentuaalinen osuus datasta on poikkeamia sisältäviä signaaleja.

Pylväskaavion lisäksi yhteenvetonäkymä sisältää yksittäisen pylvään, jossa esitetään poikkeamien kokonaismäärä koko datassa. Tämä pylväs on aina samankorkuinen riippumatta poikkeamien määrästä. Pylväs on jaettu poikkeamien määrien suhteessa erivärisiin osiin, aivan kuten pylväskaavion pylväät. Tämän lisäksi kuhunkin osaan on merkitty sitä vastaavan poikkeamatyyppin lyhenne ja kyseisen tyyppisten poikkeamien koko-



naismäärä signaaleissa yhteensä. Pylvään alapuolella on kaikenlaisien poikkeamien kokonaismäärä.

#### 4.4.2 Datanäkymä

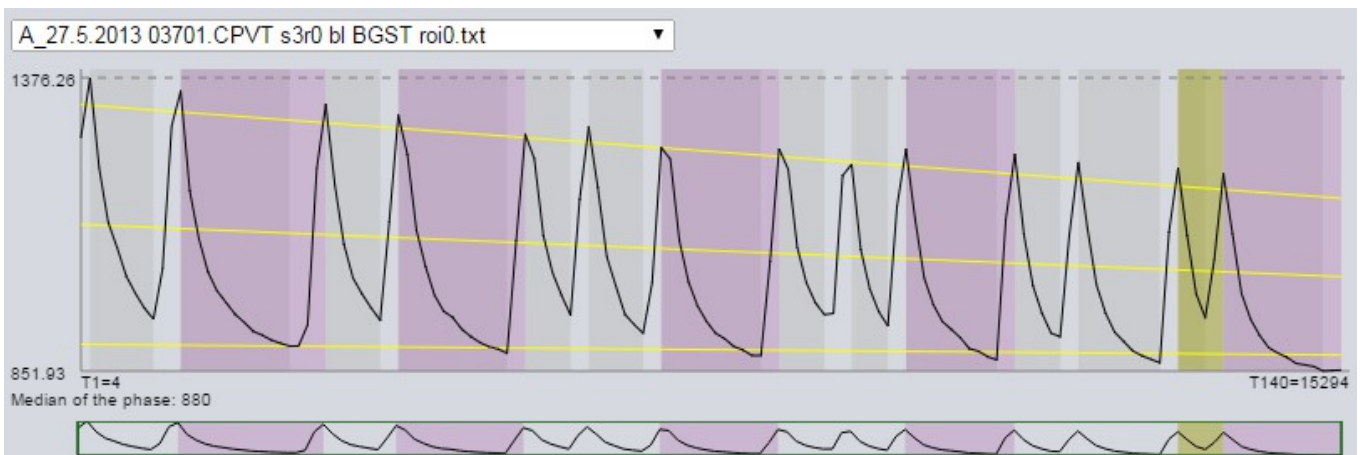
Datanäkymä (Kuva 27) on hyvin yksinkertainen. Se koostuu kahdesta listasta, joista toisessa on normaalisti luokitellut signaalit ja toisessa puolestaan epänormaalisti luokitellut. Tämä näkymä oli käyttäjiltä tullut toive. Sitä hyödynnetään sopivien parametriarvojen hakemisessa. Käyttäjä voi ensin luokitella itse manuaalisesti joukon signaaleja ja sen jälkeen tarkistaa tästä näkymästä, onko ohjelma luokitellut signaalit samalla tavalla käytetyillä parametrien arvoilla.

Regular	Abnormal
R_4.5.2012_05503.CPVT_s1r1,5_bl_BGST_roi5.txt	A_4.5.2012_05503.CPVT_s2r1,2,3,4,5_bl_BGST_roi2.txt
R_4.5.2012_05503.CPVT_s2r1,2,3,4,5_bl_BGST_roi5.txt	A_24.04.12_05503.CPVT_s3r0_bl_BGST_roi0.txt
R_24.04.12_05503.CPVT_s2r0r1_bl_BGST_roi0.txt	A_24.04.12_05503.CPVT_s5r2_bl_BGST_roi2.txt
R_24.04.12_05503.CPVT_s2r0r1_bl_BGST_roi1.txt	A_150713_05503.CPVT_s02_bl_Ratio #1_BGST_ROI1.txt
R_24.04.12_05503.CPVT_s4r0_bl_BGST_roi0.txt	A_150713_05503.CPVT_s01_bl_Ratio #1_BGST_ROI9.txt
R_24.04.12_05503.CPVT_s4r2_bl_BGST_roi2.txt	A_150713_05503.CPVT_s02_bl_Ratio #1_BGST_ROI2.txt
R_24.04.12_05503.CPVT_s6r0_bl_BGST_roi0.txt	A_160713_05503.CPVT_s01_bl_Ratio #2_BGST_ROI2.txt
R_24.4.12_05503.CPVT_s1_bl_BGST_roi0.txt	A_160713_05503.CPVT_s03_bl_r7_X_Ratio #3.txt
R_24.4.12_05503.CPVT_s1_bl_BGST_roi1.txt	A_160713_05503.CPVT_s03_bl_X_Ratio #3_roi1.txt
R_110712_05503.CPVT_s2r1_bl_BGST_roi_0.txt	A_160713_05503.CPVT_s03_bl_X_Ratio #3_roi2.txt
R_150713_05503.CPVT_s01_bl_Ratio #1_BGST_ROI1.txt	A_160713_05503.CPVT_s03_bl_X_Ratio #3_roi7.txt
R_160713_05503.CPVT_s07_bl_X_Ratio #3_roi14.txt	A_160713_05503.CPVT_s07_bl_Ratio #2_BGST_ROI1.txt
R_160713_05503.CPVT_s11_bl_Ratio #1_BGST_ROI5.txt	
R_160713_05503.CPVT_s11_bl_X_Ratio #3_roi5.txt	

Kuva 27: Datanäkymä

#### 4.4.3 Analyysinäkymä

Analyysinäkymä (Kuva 28) koostuu kolmesta osasta: pudotusvalikosta ja kahdesta signaalikäyrän kuvasta. Pudotusvalikossa on listattuna kaikki datan sisältämät signaalit, normaalit ja epänormaalit. Kun pudotusvalikon avaa, poikkeamia sisältävien signaalien



Kuva 28: Analyysinäkymä

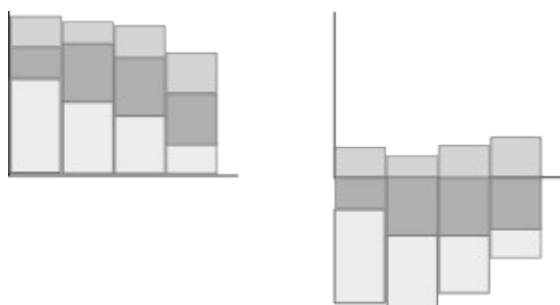
nimet ovat punaisella tekstillä ja normaalit signaalit mustalla tekstillä. Molemmat signaalikäyrät esittävät samaa pudotusvalikosta valittua signaalia.

Signaalikäyrät on sijoitettu allekkain ja alempi niistä on hyvin matala. Se on signaalin yleiskuva ja liittyy vuorovaikutukseen. Sen merkitystä kuvataan tarkemmin myöhemmin kohdassa 4.5. Molemmissa kuvissa näkyy itse käyrä sekä poikkeamat merkitynä taustavärein. Ylemmässä kuvassa esitetään myös regressiosuorat ja omalla taustaväriällään nousevat ja laskevat kohdat signaalissa. Laskevien ja nousevien kohtien esittäminen on tärkeää, koska käyttäjä voi säätää kohinan määrää. Kohina on mittaukseen liittyvää epävarmuutta, jonka visualisoinnin tärkeyttä myös Ware [2012, 28] painottaa. Tekstuaalisesti on lisäksi esitetty signaalin pienin ja suurin arvo sekä ensimmäisen ja viimeisen mittauksen järjestysnumero ja mittausaika.

#### 4.5 Toiminnot ja vuorovaikutus

Näkymissä on sekä näkymän sisäisiä toimintoja että näkymien kesken linkitettyjä toimintoja. Sisäiset vaikuttavat ainoastaan kyseiseen näkymään itseensä. Linkitetyt toiminnot puolestaan vaikuttavat joko johonkin toiseen tai kaikkiin näkymiin. Alakohdassa 3.2.2 käsitellyistä vuorovaikutustekniikoista sovelluksessa on hyödynnetty valintaa, järjestämistä, suodatusta, navigointia (zoomaus) sekä yhdistämistä, joka tässä liittyy valintojen koordinointiin näkymien välillä.

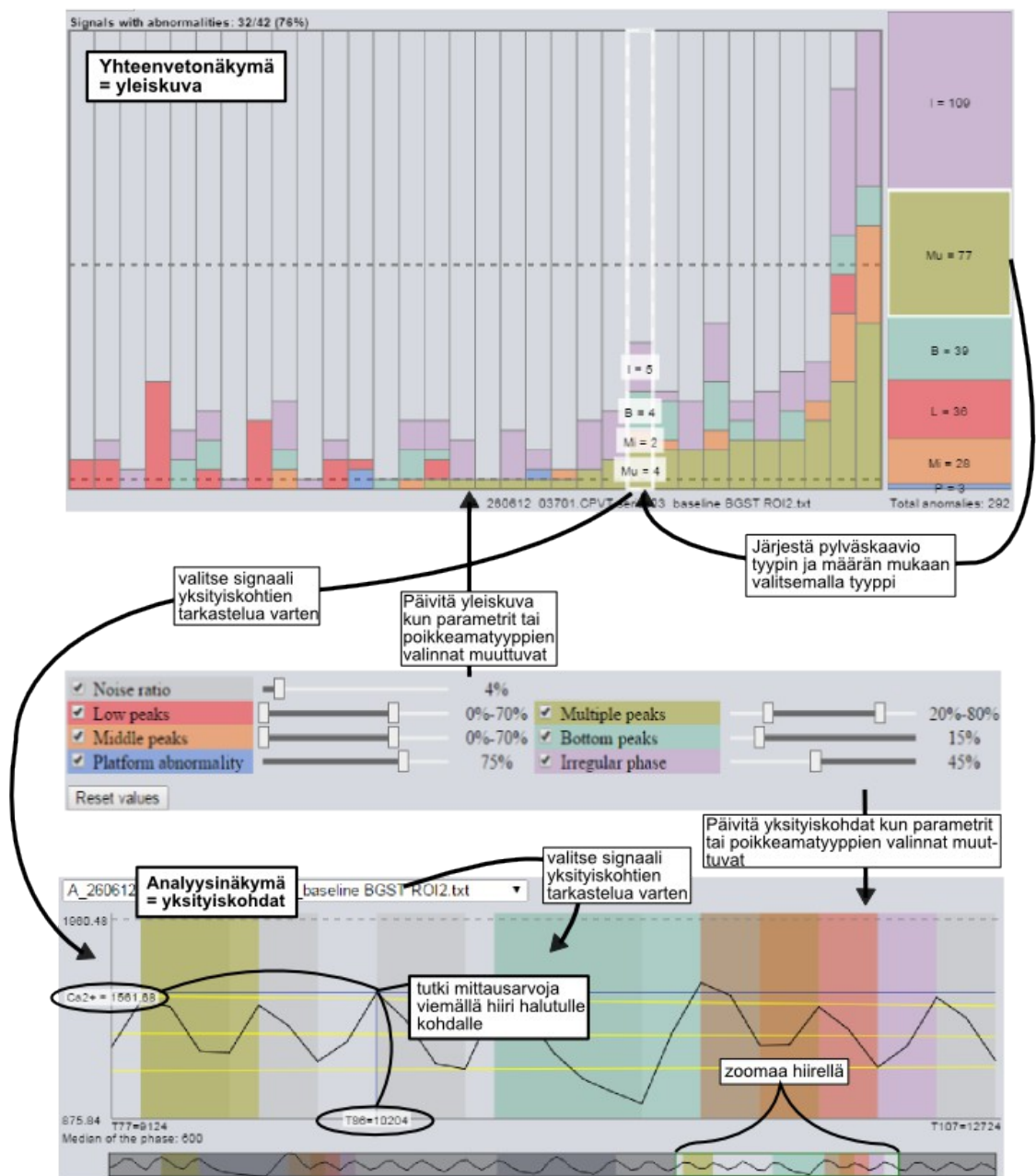
Yhteenvetonäkymällä on yksi sisäinen toiminto. Sen erillisestä pylväästä voi valita haluamansa poikkeamatyyppin, jolloin yksittäisten signaalien pylväskaavio järjestetään kyseisen poikkeamatyyppin määrien mukaan, kasvavasti vasemmalta oikealle (Kuva 30, Yhteenvetonäkymä). Lisäksi valittu poikkeamatyyppi piirretään pylväsesityksessä alimmaksi, jotta kyseisen tyyppin jakauma signaalien kesken on helppo nähdä. Muuten pystysuuntainen piirrosjärjestys on sama kuin erillisessä pylväässä. Toinen vaihtoehto olisi käyttää Dixin ja Ellisin [1998] tekniikkaa, jossa pylväät pudotetaan siten, että valittu tyyppi on tasattu diagrammin alkuperäiseen alareunaan ja niiden alapuoliset osat piirretään diagrammin alle. Esimerkki tästä on kuvassa 29. Tekniikan haittapuolena on kuitenkin tilan käyttö: pahimmillaan yhteenvetonäkymä veisi lähes kaksinkertaisen tilan pystysuunnassa.



Kuva 29: Esimerkki Dixin ja Ellisin [1998] tekniikasta

Yhteenvedonäkymä on myös linkitetty analyysinäkymään. Yhteenvedonäkymästä voi valita yksittäisen signaalin pylvään, jolloin se korostetaan erilaisella reunaviivalla ja pylvään päällä näkyy poikkeamien määrät myös numeerisesti. Kyseinen signaali tulee samalla valituksi myös analyysinäkymään (Kuva 30).

Analyysinäkymä puolestaan on linkitetty yhteenvedoon. Signaalin voi valita analysoitavaksi myös analyysinäkymän pudotusvalikosta. Tällöin valittu signaali korostetaan yhteenvedosta vastaavasti kuin jos valinta olisi tehty yhteenvedonäkymästä.



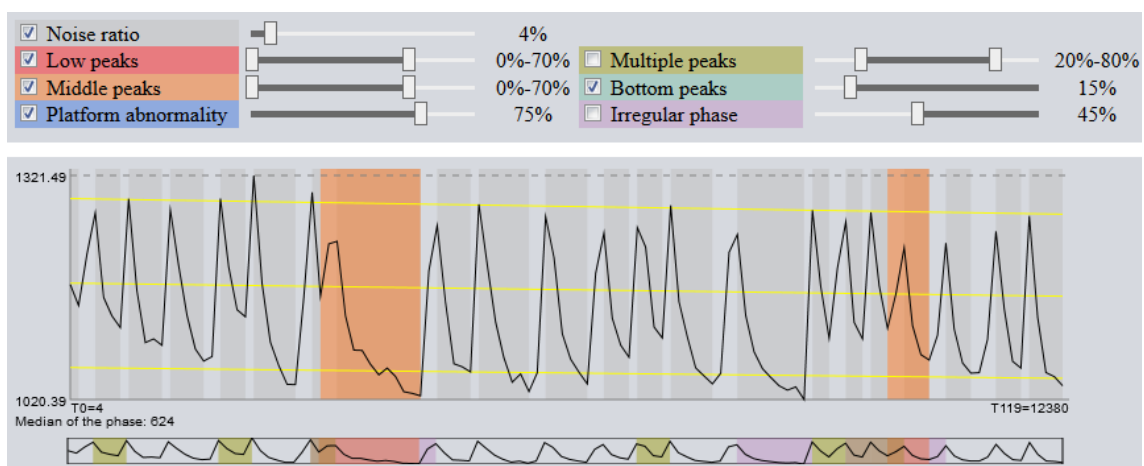
Kuva 30: Vuorovaikutus

Analyysinäkymän sisäiset toiminnot liittyvät signaalin tarkempaan tutkimiseen. Kun vie hiiren kursorin signaalikäyrän päälle, esitetään kursoria lähin mittaustulos vaakasuunnassa (Kuva 30, Analyysinäkymä). Tuloksesta esitetään sen kohta käyrällä pysty- ja vaakasuorien viivojen leikkauskohdassa, sekä mittauksen järjestysnumero, mittausaika ja arvo. Analyysinäkymässä voi myös zoomata, mikäli haluaa tutkia jotakin käyrän kohtaa tarkemmin. Tällöinkin signaalin yleiskuvassa näkyy koko signaali, ja ainoastaan ylempi signaali zoomautuu. Yleiskuvassa harmaannutetaan se osa signaalista, joka ei ole näkyvässä zoomauksen jälkeen (Kuva 30, Analyysinäkymä). Siten käyttäjällä säilyy konteksti ja ymmärrys siitä, mikä kohta signaalista on tarkasteltavana.

Käyttäjän on mahdollista määrittää, montako signaalikuvaajaa analyysinäkymässä esitetään. Jokaiseen kuvaajaan liittyy aiemmin mainitut komponentit: pudotusvalikko ja kaksi signaalikäyrää. Nämä kuvaajat on tarkoitettu mahdollistamaan signaalien vertailu. Kuvaajat esitetään allekkain, jotta ne mahtuvat näkyviin yhtä aikaa – jos käyttäjä joutuisi vierittämään sivua, olisi se vertailun kannalta (lähes) sama kuin jos käyttäjä joutuisi vaihtamaan signaalia yhdessä näkymässä. Käyttäjä joutuisi painamaan mieleensä, miltä käyrä näytti ja tekemään vertailun päässään. Yhtäaikainen signaalien näkeminen on kognitiivisesti vähemmän kuormittavaa. Toki signaalit mahtuisivat näytölle myös vierrekkäin, mutta vertailu on helpompaa allekkain: silmille riittää pienempi liike, koska näkymät ovat pystysuunnassa pienempiä kuin vaakasuunnassa.

Parametrinäkymä on linkitetty kaikkiin näkymiin, mutta vaikutukset eri näkymiin riippuvat toiminnosta. Parametrinäkymässä on sekä liukusäätimiä että valintaruutuja, jotka mahdollistavat *dynaamiset kyselyt* (dynamic query): muutokset heijastuvat reaaliaikaisesti visualisointeihin. Liukusäätimien arvojen muuttuminen vaikuttaa kaikkiin näkymiin. Koska arvojen muutokset vaikuttavat poikkeamiin, yhteenveto- ja analyysinäkymissä päivittyä tarvittaessa poikkeamien määrät ja kohdat. Arvojen muutos voi johtaa myös siihen, että poikkeamat katoavat epänormaaliksi luokitellusta signaalista ja siitä tuleekin normaali tai päinvastoin eli normaaliksi luokiteltuun signaaliin ilmestyy poikkeamia, jolloin siitä tulee epänormaali. Tällöin normaaliksi muuttuneet signaalit poistetaan yhteenvedosta ja epänormaaliksi muuttuneet lisätään siihen. Myös datanäkymässä signaalit siirretään reaaliaikaisesti oikeisiin listoihin.

Parametrinäkymän liukusäätimiin liitettyjen valintaruutujen avulla voi suodattaa poikkeamatyyppejä sekä yhteenveto- että analyysinäkymässä. Ainoastaan valitut poikkeamatyypit piirretään näkyviin. Kokonaisuhteenvetopylväässä kaikki tyypit kuitenkin esitetään aina huolimatta valinnoista. Samoin tehdään analyysinäkymän yleiskuvassa eli matalammassa signaalikuvassa (Kuva 31).



Kuva 31: Näkyvien parametrivalintojen vaikutukset analyysinäkymään: kaikki poikkeamat esitetään alemmassa signaalikäyrässä, mutta ylemmässä ainoastaan parametrinäkymässä valitut

## 5. Pohdinta

### 5.1 Käytettävyydestä

Ohjelma on ollut koekäytössä kahdella kalsiumsignaaleja työssään tutkivalla henkilöllä. Heiltä saadun palautteen perusteella ohjelma on lupaava, mutta joitakin kehittämistä vaatia seikkoja on. Ohjelmalle ei ole tehty käytettävyystudkimusta ja sellainen voisi olla syytä tehdä.

Suurin käytettävyysohjelma nykyisessä versiossa vaikuttaisi olevan päällekkäiset poikkeamat. Poikkeamien tunnistusmekanismi saattaa tunnistaa signaalin joissakin ajankohdissa useita poikkeamia. Analyysinäkymän signaalikäyrässä sellaiset kohdat näkyvät harmaina tai ruskeina. Tämä johtuu siitä, että väreissä on käytetty läpinäkyvyyttä, joten päällekkäin piirrettävät värit sekoittuvat.

Harmaus on ongelma, koska siitä ei pysty tunnistamaan, mitä poikkeamia kyseisellä kohdalla on (Kuva 32). Tunnistaminen vaatii valintaruutujen valintojen vaihtelua päälle ja pois, jotta saa oikeat värit näkyviin. Läpinäkymättömien värien käyttäminen ei luonnollisestikaan korjaisi ongelmaa, koska silloin ei näkyisi, että kohdassa on useammanlaisia poikkeamia, vaan ainoastaan viimeiseksi piirretty väri näkyisi.

Ongelman ratkaisu vaatisi uudenlaista esitystapaa. Esimerkiksi voisi kokeilla kohdan taustanelikulmion värittämistä vaakasuorin raidoin niiden poikkeamien väreillä, joita kohdassa on tunnistettu. Vielä ei kuitenkaan olla päästy lopulliseen tulokseen siitä, pitäisikö päällekkäisiä poikkeamia edes havaita vai pitäisikö poikkeamien olla toisensa pois-sulkevia.

Myös yhteenvetonäkymään liittyy kaksi potentiaalista käytettävyysohjelmaa. Käyttäjää haastateltaessa kävi ilmi, ettei toinen ollut lainkaan huomannut, että erillistä kokonaisyhteenvetopylvästä klikkaamalla pystyi muuttamaan signaalien järjestystä yhteenvedossa. Toisaalta hän kertoi, ettei ollut vielä käyttänyt näkymää kovinkaan paljon. Mahdollisessa käytettävyystudkimuksessa olisikin hyvä selvittää, onko tämä yleisempikin ongelma ja pitäisikö pylvään ulkoasua kehittää, jotta se näyttäisi enemmän vuorovaikutteiselta.

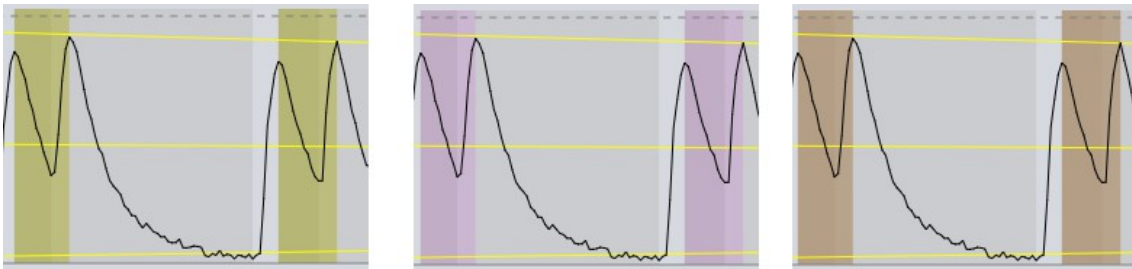
Toinen käytettävyysohjelma yhteenvetonäkymässä liittyy parametrien valintoihin. Kun vain osa parametreista on valittuna, näkyy yhteenvedossa sellaisiakin signaaleja, joissa ei ole yhtään valittujen tyyppistä poikkeamaa (Kuva 33), vaan ainoastaan valitsemattomia poikkeamia. Tätä toinen käyttäjä piti virheellisenä toimintana.

Tällä hetkellä poikkeamatyyppien valinta vaikuttaa ainoastaan analyysinäkymän signaalin yksityiskohtaiseen kuvaan. Sen alla olevassa pienemmässä signaalin kuvaajas- sa kaikki poikkeamatyyppit ovat aina tarkoituksella näkyvissä, jotta siitä saa aina yleisku-

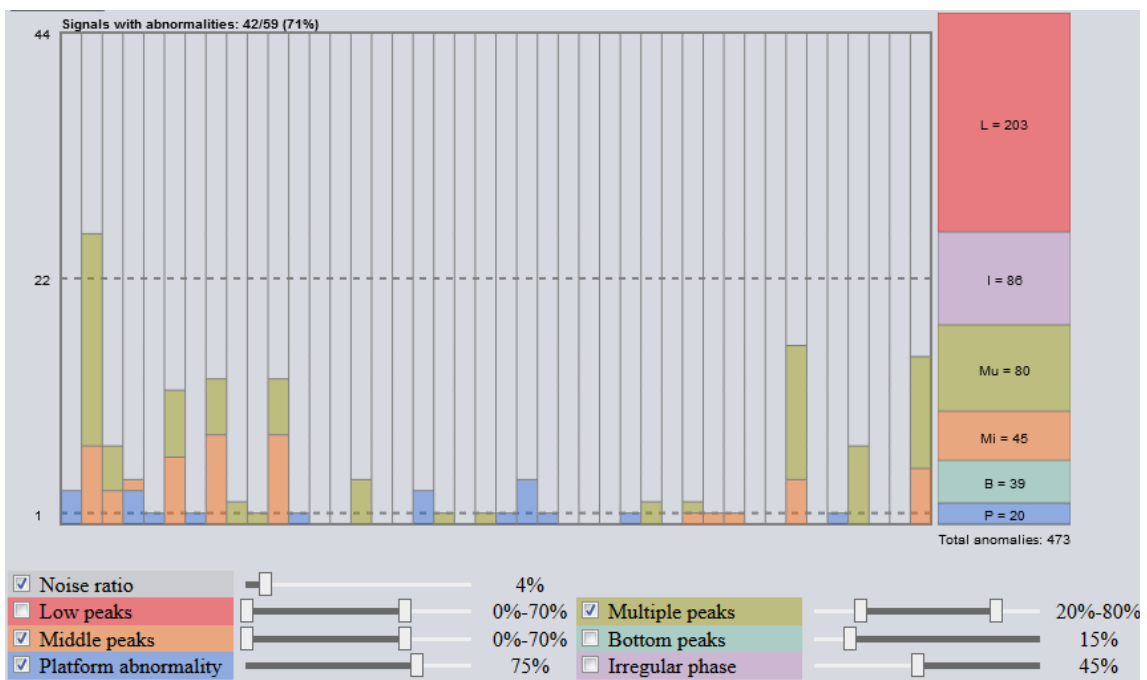
van signaalista, vaikka joitakin poikkeamatyyppjeä ei olisi valittuna tai signaalia olisi zoomattu. Tämä toiminnallisuus on perusteltua pitää nykyisenlaisena.

Voi olla ongelmallista vain poistaa yhteenvedosta ne signaalit, joissa ei juuri valittuna olevia poikkeamia ole havaittu. Pitäisikö signaali tällöin käsittää normaaliksi? Silloin pitäisi poistaa poikkeamatyyppi myös erillisestä yhteenvetopylväästä ja muuttaa yhteenvedossa kerrottujen poikkeavien signaalien lukumäärän ja osuuden sekä poikkeamien kokonaismäärän arvoja. Lisäksi pitäisi muuttaa signaalien nimien väriä analyysinäkymän pudotusvalikossa ja siirtää signaalit epänormaaleista normaaleihin datanäkymässä.

Vaihtoehtoinen tapa korjata tilanne voisi olla jättää näkymät muuten ennalleen, mutta toivotun mukaisesti poistaa yhteenvedosta signaaleja ja lisätä tiedot siitä, paljonko on signaaleja, joissa on valittuja poikkeamia sekä paljonko yhteensä valittuja poikkeamia on.



Kuva 32: Pällekkäisten poikkeamien ongelma. Pätkä samasta signaalista analyysinäkymässä, kun eri poikkeamatyyppjeä valittuna. Vasemmalla valittuna vain monihuippuiset piikit, keskellä epäsäännölliset vaiheet ja oikealla molemmat.



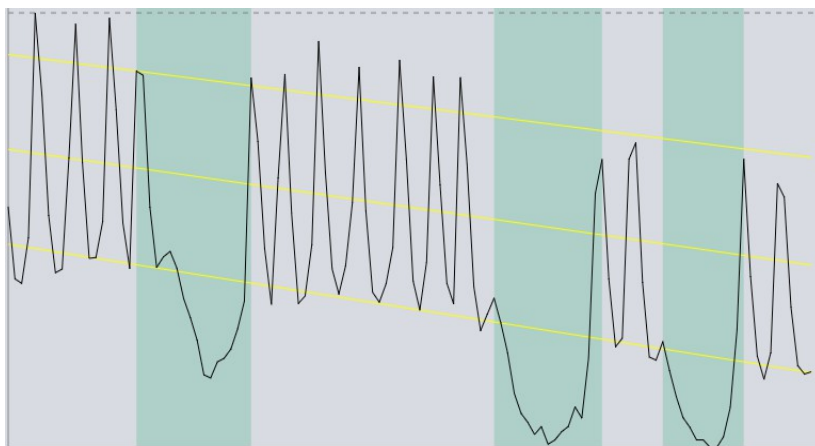
Kuva 33: Yhteenvedossa signaaleja, joissa ei ole valittujen tyyppien poikkeamia

## 5.2 Jatkokehitys

Ohjelma on ollut vasta alustavassa käytössä, joten erilaisia kehitystarpeita ilmenee vielä myöhemminkin, mutta ainakin toistaiseksi kriittisin kehitystarve käytettävyysongelmien korjaamisen lisäksi koskee poikkeamien tunnistusta ja yhteenvetönäkymää.

### 5.2.1 Poikkeamat

Poikkeamiin liittyy useita asioita, joita täytyy vielä kehittää. Asiantuntijat ovat tunnistaneet ainakin yhden poikkeamatyyppin, jonka ohjelmaan voisi lisätä. Tyyppi on *oskillaatio*, joka tarkoittaa sitä, että signaali jää jossakin vaiheessa vaihtelemaan kahden arvon välillä. Oskillaatio on esitetty kuvassa 34, jossa ohjelma on tunnistanut kolme alapiikkiä. Asiantuntija kuitenkin arvioi, että nuo alapiikit ovat signaalin normaaleja pohja-arvoja ja niiden sijasta pitäisikin tunnistaa niiden väliset alueet oskillaatioksi.



Kuva 34: Oskillaatio

Oskillaation tunnistaminen voisi ainakin osittain perustua monihuippuisen piikin tunnistamiseen. Asiantuntijan mukaan monihuippuinen piikki on oskillaatio, jos piikkejä on enemmän kuin kaksi. Oskillaation erottaminen monihuippuisesta piikistä ei kuitenkaan ratkaise ongelmaa, joka näkyy kuvassa 34. Kuvan tapauksessa regressiokäyrä tulisikin laskea niistä kolmesta alimmasta pohjasta, joissa nyt on havaittu alapiikki. Tällöin ohjelma tunnistaisi pohjat normaaleiksi ja välit todennäköisesti monihuippuisiksi piikeiksi.

Vaikka oskillaatio erotettaisiin monihuippuisesta piikistä piikkien määrästä, voisi kuvan 34 tapaus jäädä ongelmalliseksi: kuvassa kahden viimeisen alapiikiksi tunnistetun välissä oleva kohta tunnistettaisiin joka tapauksessa monihuippuiseksi piikiksi, vaikka voisi olla luontevampaa tulkita sekin oskillaationa tässä tapauksessa.

Kuvan tapausta ei kuitenkaan voi suin päin pyrkiä korjaamaan muuttamalla regressiokäyrien laskentaa, koska niiden laskentatapa vaikuttaa kaikkien poikkeamien tunnis-



tamiseen. Voi osoittautua hyvinkin vaikeaksi erottaa tapaukset, joissa pitäisi tunnistaa alapiikkejä, tapauksista, joissa on kyse oskillaatiosta. Voisi pohtia, olisiko jokin osuus alapiikkejä kaikista piikeistä kynnsarvo, joka kääntäisi tunnistamisen: alapiikkien kohdat olisivatkin normaaleja ja niiden välit oskillaatiota.

Kehitettävää on mahdollisesti myös parametrien säätämisen suhteen. Tämän hetkiset ohjelman oletusarvot eivät ole käyttäjäpalutteen mukaan toimivia. Toki oletusarvojen löytäminen on asiantuntijoiden harteilla. Toisaalta kun poikkeamien laskentaa kehitetään edelleen, vaikuttaa sekin siihen, mitä arvojen pitäisi olla. Nykyisellä laskentatavalla löytyvät toimivat arvot eivät välttämättä ole enää hyvät, jos laskentatapaa muutetaan.

Käyttäjiltä on lisäksi tullut toive mahdollisuudesta tallentaa parametriasetuksia. Sen voi toteuttaa useilla tavoin. Yksi vaihtoehto voisi olla W3C:n määrittelemä Web Storage -rajapinta, joka on tarkoitettu vaihtoehdoksi evästeille tiedon tallentamiseen selaimen puolelle. Sitä voisi käyttää myös automaattiseen arvojen tallentamiseen joka kerta, kun ne muuttuvat, ja tällöin sivun avatessa edelliset arvot olisivat valmiina. Käyttäjien toiveena oli kuitenkin ennen kaikkea tallentaa arvot halutessaan.

Varmempaa voisi olla tallentaa ne omiin tiedostoihinsa, erityisesti jos tarve on tallentaa useampi parametriarvojen yhdistelmä eri käyttötarkoituksia varten. Tiedostoon kirjoittaminen ei kuitenkaan turvallisuussyistä onnistu suoraan JavaScript-koodista. On kuitenkin mahdollista generoida koodissa sisältöä, jonka käyttäjä voi ladata ja tallentaa. Se vaatii tietysti käyttäjältä lisätoimia, mutta toisaalta vaikka tiedot tallennettaisiin Web Storaagen avulla, täytyisi käyttäjää pyytää nimeämään arvoyhdistelmän, mikäli niitä voisi tallentaa useampia.

Mikäli sivusto taas laitetaan vakituisesti jollekin palvelimelle käytettäväksi, ja tallennettuja parametriarvoja halutaan jakaa muiden käytettäväksi, olisi tietysti mahdollista toteuttaa myös tallennus palvelinpuolelle. Tämä olisi työläin vaihtoehto, koska nykyisessä versiossa ei ole palvelinpuolta lainkaan ja se pitäisi toteuttaa. Toki on olemassa lukuisia ohjelmistokehyksiä, jotka tekevät asiakas-palvelin-kokonaisuuden rakentamisesta varsin helppoa.

Käyttäjiltä on tullut myös toive muuttaa ohjelman tekemää luokittelua manuaalisesti. Ensisijaisesti pitäisi kuitenkin pyrkiä parantamaan ohjelman automaattista luokittelua. Kun tavoitteena on objektiivinen luokittelu, manuaalinen tulosten muokkaus olisi sitä vastaan. Toki käyttäjät halusivatkin, että ohjelma esittäisi jotenkin ne kohdat, joissa käyttäjä on muuttanut luokittelua.

Mikäli tullaan siihen tulokseen, että luokittelun manuaalinen muuttaminen toteutetaan, vaikuttaa se kaikkiin näkymiin: jokaisessa täytyy esittää jotenkin, että luokittelua on muutettu. Mikäli näyttää siltä, että riittää poikkeamien poistaminen, kohtuullisen selkeä tapa voisi olla esittää poistetut poikkeamat analyysinäkymässä katkoviivoin reunus-

tettuina ja poikkeaman väri normaalia vaaleampana. Yhteenvedonäkymässä voisi esittää vastaavalla tavalla poistettujen poikkeamien osuuden.

Poistamistoiminto itsessään täytyisi olla vain analyysinäkymässä, joka on ainut näkymä, josta käyttäjä voi nähdä, miltä poikkeamaksi luokiteltu kohta signaalissa näyttää. Esimerkiksi poikkeamaa klikatessa voisi käyttäjältä kysyä, haluaako hän poistaa kyseisen poikkeaman – tai jos kohdalla on useampia poikkeamia, käyttäjälle voisi tarjota jonkinlaisen listan, josta hän voi valita poistettavat poikkeamat.

Jos käyttäjällä on tarve myös lisätä poikkeamia, luonnollinen tapa olisi halutun kohdan maalaaminen analyysinäkymässä. Lisätyille poikkeamille täytyisi lisäksi kehittää oma esitystapansa, joka poikkeaisi ohjelman havaitsemista ja niistä, jotka käyttäjä on poistanut.

Poikkeamien poistaminen tai lisääminen voi vaikuttaa myös datanäkymään, jos käy niin, että tehdyt muutokset tekevät epänormaaliksi luokitellusta signaalista normaalin tai päinvastoin. Kummassa listassa signaali tällöin pitäisi esittää? Vai molemmissa, kenties värikoodattuna?

Kaiken kaikkiaan poikkeamien manuaalinen muuttaminen on ongelmallista. Paitsi että se sotii objektiivisen luokittelun tavoitetta vastaan, sen toteuttaminen myös potentiaalisesti heikentää visuaalista luettavuutta, erityisesti jos tarvitaan sekä poikkeamien poistaminen että lisääminen. Se saattaisi aiheuttaa myös muita lisätoimintotarpeita, esimerkiksi tarvittaessa poistettujen poikkeamien piilottaminen yhteenvedossa.

### **5.2.2 Yhteenvedonäkymä**

Yhteenvedonäkymä on tässä vaiheessa ollut varsin vähäisen käytön kohteena, eikä vielä ole selvää, mitä kaikkea siinä tarvitsisi esittää. Nyt siinä esitetään ainoastaan poikkeamien lukumäärät, mutta muitakin vaihtoehtoja on. Esimerkiksi aiemmassa, Java-pohjaisessa visualisoinnissa [Siirtola et al., 2014] yhteenvedossa on poikkeamien määrän lisäksi mahdollista tarkastella poikkeamia huippua kohti ja poikkeamien suhteellista osuutta signaalin mittausajasta. Näitä tapoja ei kuitenkaan nähty tarpeelliseksi toteuttaa tässä esiteltyyn selainversioon.

Käyttäjät itsekin arvelevat, että toiveita näkymän suhteen tulee vielä ilmenemään, kun ohjelman käytössä ja signaalien tutkimisessa on edetty. Voi myös olla mahdollista, että tulevat tarpeet vaativat aivan uudenlaista esitystapaa, eikä vain suureita y-akselilla.

### **5.2.3 Signaalien vertailu**

Signaalien vertailu on käyttäjien toivoma toiminto, joka on tarpeen lääkevasteen tutkimisessa. Toiminnon nykyinen toteutus on vain mahdollisuus avata useampia analyysinäkymiä. Signaalien vertaileminen on kuitenkin toimintona erillinen muusta toiminnallisuudesta. Se ei varsinaisesti liity ohjelman ydintoimintoon, signaalien luokitteluun. Voisi olla syytä harkita vertailun erottamista kokonaan omaksi näkymäkseen.

Nykytoteutuksessa eri signaalit näkyvät erillisissä, päällekkäisissä näkymissä. Toinen vaihtoehto olisi piirtää signaalit samaan kuvaan. Silloin tosin signaalien vertailu voisi olla hankalaa tapauksissa, joissa mittaukset eivät ala signaalin samasta vaiheesta. Toki olisi mahdollista siirtää signaaleja vaakasuunnassa siten, että esimerkiksi vertailtavien signaalien ensimmäiset huiput osuisivat kohdilleen.

Vertailutoiminnon käyttötarpeita ja -tapoja voisi selvittää tarkemmin. Olisiko mahdollisesti tarvetta vertailla myös signaalijoukkoja, eikä vain yksittäisiä signaaleja? Tätä nykyinen versio ohjelmasta ei tue.

### 5.3 Tavoitteiden täyttyminen

Teknisinä vaatimuksina oli alustariippumattomuus ja ettei ohjelmaa tarvitsisi asentaa. Verkkosivu vastaa näihin vaatimukseen. Verkkosivua ei luonnollisesti tarvitse asentaa – sen käyttö onnistuu joko selaimen www-palvelimelta hakemana, tai jos syystä tai toisesta ohjelmaa ei haluta laittaa verkkoon saataville, onnistuu käyttö myös lokaalisti kopioimalla tiedostot omalle koneelle.

Verkkosivu on alustariippumaton. Verkkosivujen toiminta ei periaatteessa ole kiinni käyttöjärjestelmästä tai laitteesta, ja niiden hyvänä puolena onkin, ettei niistä tarvitse erillisiä versioita. Sivujen varsinainen alusta onkin selain, ja niissä voi olla eroja, jotka täytyy huomioida sivujen toteutuksessa. Koska toteutuksessa käytettiin HTML:n versiota 5, ei sivu toimi vanhoilla selaimilla, mutta ei ole kovin realistista, että kohderyhmällä olisi enää niin vanhoja selaimia käytössään.

Suurimmaksi osaksi toteutuksessa käytettiin ECMAScriptin ja W3C:n standardien mukaisia ominaisuuksia. Lisäksi niitä selaimen ominaisuuksia varten, joita tarvittiin, mutta joita ei kattanut mikään standardi, käytettiin jQuery-kirjastoa. Näin ollen sivun pitäisi toimia kaikilla uusilla selaimilla.

Tämän hetkiseen toteutukseen liittyy kuitenkin yksi alustaan liittyvä puute: kosketusnäytöt. HTML5:n ja jQuery:n käyttöliittymäkomponenttien pitäisi kyllä toimia kosketusnäytöillä, mutta ohjelmaa varten tehdyt visualisointikomponentit toimivat toistaiseksi vain hiirellä. Koekäyttäjillä ei ole ollut käytössään kosketusnäyttöjä, joten niitä ei ole ainakaan vielä tarvinnut tukea. Ohjelmassa siis käsitellään tällä hetkellä ainoastaan W3C:n Document Object Model Events -standardissa määriteltyjä hiiritapahtumia ja kosketusnäytöt vaatisivat erillisessä Touch Events -standardissa määriteltyjen kosketustapahtumien käsittelyä.

Toiminnallisina vaatimuksina oli signaalien luokittelu normaaleihin ja epänormaaleihin, poikkeamien tunnistaminen sekä signaalien vertailu. Nämä ominaisuudet on toteutettu, mutta vaativat pientä käytettävyyden hiomista ja mahdollisesti muuta jatkokehittelyä, mitä on selvitetty tarkemmin muualla tässä luvussa.

Tavoitteena oli objektiivinen menetelmä, jonka avulla eri tutkijat voivat saada samat tulokset. Samoja parametriarvoja käyttämällä ohjelma luokittelee samat signaalit aina samoin, joten siltä osin vaatimukset toteutuvat.

Luokittelusta haluttiin myös nopeampaa ja yksityiskohtaisempaa. Luokittelun nopeus ohjelmalla riippuu eniten sopivien parametriarvojen löytymisestä. Jos keksitään parametriarvot, jotka toimivat useimmissa tapauksissa, luokittelu on todennäköisesti nopeampaa ohjelmalla. Toki sitä voisi vielä tutkia.

Luonnollisestikin luokittelun pitäisi olla myös oikeellista, muuten kaikki muut tavoitteet ovat tietenkin turhia. Olisikin erityisen tärkeää tutkia, kuinka hyvin ohjelma luokittelee signaaleja verrattuna asiantuntijaan. Käyttäjiltä saatujen alustavien arvioiden perusteella luokittelu ohjelmalla on ollut tarkempaa kuin kohdassa 2.3 kuvatuilla Juholan ja muiden [2014] tutkimilla tiedonlouhintamenetelmillä, joten ohjelma on hyvin luopaava.

## 6. Yhteenveto

Tässä tutkielmassa kuvattiin sydämen kalsiumsignaalia ja tarvetta sen poikkeamien luokitteluun. Tässä myös esiteltiin aiempia ratkaisuja, joita löytyi vain yksi: tiedonlouhinta-algoritmien hyödyntäminen. Tässä esitettiin visuaaliseen analytiikkaan perustuvaa lähestymistapaa, ja esiteltiin siihen perustuva ratkaisu.

Visuaalinen analytiikka yhdistää erilaisia tietokoneeseen ja ihmisen tiedonkäsittelyyn liittyviä aloja. Tarkoituksena on ihmisen ja koneen yhteistyö, jossa käytetään hyväksi molempien vahvuuksia, koneiden laskentatehoa ja ihmisen tehokasta visuaalisten havaintojen käsittelyä.

Ratkaisuksi visuaalisen analytiikan pohjalta syntyi vuorovaikutteinen visualisointi, jossa käyttäjä näkee reaaliaikaisesti parametrien säätämisen vaikutukset laskennan tulokseen. Sovellus toteutettiin verkkosivuna, jotta se on alustariippumaton eikä vaadi asentamista.

Seuraavaksi pitäisi tehdä vertailututkimus siitä, miten hyvin sovelluksen luokittelut vastaavat asiantuntijan tekemiä. Käyttäjien itsensä tekemän alustavan vertailun tulokset vaikuttavat lupaavilta: sovellus luokittelee signaaleja tarkemmin kuin aiemmin tutkitut tiedonlouhinta-algoritmit.

Mielenkiintoista on myös nähdä, löytyvätkö yleiset parametriarvot, jotka toimivat eri signaalijoukoille. Tällöin visualisointi ei periaatteessa enää olisi välttämätön. Toisaalta ilman visualisointia käyttäjä ei voi mitenkään varmistua, että yleensä käytetyt arvot toimivat myös uudelle signaalijoukolle, joten vakioarvoillakin visualisoinnilla on arvonsa.

## Viiteluettelo

- [Aigner et al., 2007] Wolfgang Aignera, Silvia Mikscha, Wolfgang Müllerb, Heidrun Schumannc, and Christian Tominski, Visualizing time-oriented data – a systematic view. *Computers & Graphics* **31** (2007), 401-409. doi: 10.1016/j.cag.2007.01.030
- [Baldonado et al., 2000] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky, Guidelines for using multiple views in information visualization. In: *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '00)*, 110-119. doi: 10.1145/345513.345271
- [Bers, 2008] Donald M. Bers, Calcium cycling and signaling in cardiac myocytes. *Annual Review of Physiology* **70** (2008), 23-49. doi: 10.1146/annurev.physiol.70.113006.100455
- [Borgo et al., 2013] R. Borgo, J. Kehler, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, Glyph-based visualization: foundations, design guidelines, techniques and applications. In: *Eurographics 2013 - State of the Art Reports*, 39-63. doi: 10.2312/conf/EG2013/stars/039-063
- [Brehmer and Munzner] Matthew Brehmer and Tamara Munzner, A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics* **19**, 12 (2013), 2376-2385. doi: 10.1109/TVCG.2013.124
- [Dix and Ellis, 1998] Alan Dix and Geoffrey Ellis, Starting simple: adding value to static visualisation through simple interaction. In: *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '98)*, 124-134. doi: 10.1145/948496.948514
- [ECMA-262] ECMA-262 Edition 5.1. 2011. ECMAScript Language Specification. Ecma International. <http://www.ecma-international.org/ecma-262/5.1/Ecma-262.pdf>
- [Gotz and Zhou, 2008] David Gotz and Michelle X. Zhou, Characterizing users' visual analytic activity for insight provenance. In: *Proceedings of IEEE Symposium on Visual Analytics Science and Technology 2008 (VAST '08)*, 123-130. doi: 10.1109/VAST.2008.4677365
- [Heer and Shneiderman, 2012] Jeffrey Heer and Ben Shneiderman, Interactive dynamics for visual analysis. *Queue – Micoprocessors* **10**, 2 (2012). doi: 10.1145/2133416.2146416
- [HTML5] HTML5 A vocabulary and associated APIs for HTML and XHTML W3C Recommendation 28 October 2014. <http://www.w3.org/TR/2014/REC-html5-20141028/>
- [Jansen and Dragicevic, 2013] Yvonne Jansen and Pierre Dragicevic, An interaction model for visualizations beyond the desktop. *IEEE Transactions on Visualization and Computer Graphics* **19**, 12 (2013), 2396-2405. doi: 10.1109/TVCG.2013.134

- [Javed and Elmqvist, 2012] Waqas Javed, and Niklas Elmqvist, Exploring the design space of composite visualization. In: *Proceedings of Pacific Visualization Symposium (PacificVis), 2012 IEEE*, 1-8. doi: 10.1109/PacificVis.2012.6183556
- [Jayapandiana et al., 2013] Catherine P. Jayapandiana, Chien-Hung Chena, Alireza Bozorgib, Samden D. Lhatoob, Guo-Qiang Zhanga, and Satya S. Sahoo, Electrophysiological signal analysis and visualization using Cloudwave for epilepsy clinical research. In: *Proceedings of the 14th World Congress on Medical and Health Informatics (MEDINFO 2013)*, 817 – 821. doi: 10.3233/978-1-61499-289-9-817
- [Juhola et al., 2014] Martti Juhola, Henry Joutsijoki, Kirsi Varpa, Jyri Saarikoski, Jyrki Rasku, Kati Iltanen, Jorma Laurikkala, Heikki Hyyrö, Jorge Avalos-Salguero, Harri Siirtola, Kirsi Penttinen, and Katriina Aalto-Setälä, On computation of calcium cycling anomalies in cardiomyocytes data. In: *Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1444-1447.
- [Kapeller et al., 2012] Christoph Kapeller, Christoph Hintermüller, and Christoph Guger, Augmented control of an avatar using an SSVEP based BCI. In: *Proceedings of the 3rd Augmented Human International Conference (AH '12)*, Article No. 27. doi:10.1145/2160125.2160152
- [Keim, 2000] Daniel A. Keim, Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics* **6**, 1 (2000), 59-78. doi: 10.1109/2945.841121
- [Kerren and Schreiber, 2012] Andreas Kerren, and Falk Schreiber, Toward the role of interaction in visual analytics. In: *Proceedings of the 2012 Winter Simulation Conference*, 1-13. doi: 10.1109/WSC.2012.6465208
- [von Landesberger et al., 2014] Tatiana von Landesberger, Sebastian Fiebig, Sebastian Bremm, Arjan Kuijper, and Dieter W. Fellner, Interaction taxonomy for tracking of user actions in visual analytics applications. In: Weidong Huang (ed.), *Handbook of Human Centric Visualization*. Springer, 2014, 653-670.
- [Lee et al., 2012] Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale, Beyond mouse and keyboard: expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics* **18**, 12 (2012), 2689-2698. doi: 10.1109/TVCG.2012.204
- [Lourenço et al., 2014] André Lourenço, Hugo Plácida Silva, Carlos Carreiras, Ana Priscila Alves, and Ana L. N. Fred, A web-based platform for biosignal visualization and annotation. *Multimedia Tools and Applications* **70** (2014), 433-460. doi: 10.1007/s11042-013-1397-9
- [Nagel et al., 2014] Till Nagel, Martina Maitan, Erik Duval, Andrew Vande Moere, Joris Klerkx, Kristian Kloeckl, and Carlo Ratti, Touching transport - a case study on visualizing metropolitan public transit on interactive tabletops. In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*, 281-288. doi: 10.1145/2598153.2598180

- [Payne et al., 2005] S. J. Payne, H. P. Arrol, S. V. Hunt, and S. P. Young, Automated classification and analysis of the calcium response of single T lymphocytes using a neural network approach. *IEEE Transactions on Neural Networks* **16**, 4 (2005), 949-958. doi: 10.1109/TNN.2005.849820
- [Pike et al., 2009] William A. Pike, John Stasko, Remco Chang, and Theresa A. O'Connell, The science of interaction. *Information Visualization* **8**, 4 (2009), 263-274. doi: doi: 10.1057/ivs.2009.22
- [Roberts, 2007] Jonathan C. Roberts, State of the art: coordinated & multiple views in exploratory visualization. In: *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '07)*, 61-71. doi: 10.1109/CMV.2007.20
- [Rowley, 2007] Jennifer Rowley, The wisdom hierarchy: representations of the DIWK hierarchy. *Journal of Information Science* **33**, 2 (2007), 163-180. doi: 10.1177/0165551506070706
- [Sacha et al., 2014] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim, Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics* **20**, 12 (2014), 1604-1613. doi: 10.1109/TVCG.2014.2346481
- [Shneiderman, 1996] Ben Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)*, 1996, 336-343. doi: 10.1109/VL.1996.545307
- [Siirtola et al., 2014] Harri Siirtola, Jorge Àvalos-Salguero, Kirsi Penttinen, Katriina Aalto-Setälä, and Martti Juhola, Interactive biosignal analysis and classification. In: *Proceedings of 2014 18th International Conference on Information Visualisation (IV 2014)*, 327-332. doi: 10.1109/IV.2014.27
- [Stevens, 1946] S. S. Stevens, On the Theory of Scales of Measurement. *Science* **103**, 2684 (1946), 677-680. doi: 10.1126/science.103.2684.677
- [Thomas and Cook, 2005] James Thomas and Kristin Cook, *Illuminating the Path*. IEEE Press, 2005.
- [Ware, 2012] Colin Ware, *Information Visualization*. Elsevier, 2012, third edition.
- [Wong and Thomas, 2004] Pak Chung Wong and Jim Thomas, Visual analytics. *IEEE Computer Graphics and Applications* **24**, 5 (2004), 20-12. doi: 10.1109/MCG.2004.39
- [Wong et al., 2014] Derek F. Wong, Lidia S. Chao, Xiaodong Zeng, Mang-I Vai, and Heng-Leong Lam, Time series for blind biosignal classification model. *Computers in Biology and Medicine* **54** (2014), 32-36. doi: 10.1016/j.compbiomed.2014.08.007
- [Yi et al., 2007] Ji Soon Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko, Toward a deeper understanding of the role of interaction information visualization. *IEEE Transactions on Visualization and Computer Graphics* **13**, 6 (2007), 1224-1231. doi: 10.1109/TVCG.2007.70515



[Zhang, 1996] Jijie Zhang, A representational analysis of relational information displays. *International Journal of Human-Computer Studies* **45** (1996), 59-74. doi: 10.1006/ijhc.1996.0042