

Állatorvostudományi Egyetem

Zoológiai Tanszék

**Az intracelluláris Ca^{2+} szint elemzésének
automatizálása erősen polarizált sejtekben**

Készítette: Fazekas Fruzsina
Biológia BSC., 2. évfolyam

Témavezető: Berekméri Eszter, PhD
Állatorvostudományi Egyetem,
Zoológiai Tanszék
Tudományos munkatárs

2023

Tartalomjegyzék

RÖVÍDÍTÉSEK JEGYZÉKE	3
1. BEVEZETÉS	4
1.1. KALCIUM A SZERVEZETBEN	4
1.2. KALCIUM SZINT MONITOROZÁSA	5
1.3. A FELVETT ADATOK ELEMZÉSE	8
2. CÉLKITŰZÉSEK	12
3. ANYAG ÉS MÓDSZER	13
3.1. DEITERS SEJTEK	13
3.2. AZ R PROGRAMOZÁSI NYELV	13
3.3. A KÓD MŰKÖDÉSE.....	14
3.3.1. <i>Start</i>	16
3.3.2. <i>Kneedle</i>	16
3.3.3. <i>Contrast maker</i>	17
3.3.4. <i>A forgatási szög számolása</i>	18
3.3.5. <i>Forgatás</i>	18
3.3.6. <i>Nyúlvány helyzetének vizsgálata</i>	20
3.3.7. <i>ROI maker</i>	21
3.3.8. <i>ROI calculator</i>	22
3.3.9. <i>Elcsúszásvizsgálat</i>	22
3.3.10. <i>Mentés</i>	23
4. EREDMÉNYEK.....	23
5. MEGBESZÉLÉS ÉS KÖVETKEZTETÉSEK	26
ÖSSZEFOGLALÓ	29
SUMMARY	30
KÖSZÖNETNYILVÁNÍTÁS	31
IRODALOMJEGYZÉK.....	32

Rövidítések jegyzéke

Ca ²⁺	kalcium ion
FRET	Fröster-féle rezonáns energia átadás
GKKI	genetikusan kódolt kalcium indikátor
IP ₃	inozitol-1,4,5-trifoszfát
ROI	vizsgálandó terület a felvételen (region of interest)
UV	ultraviola sugárzás

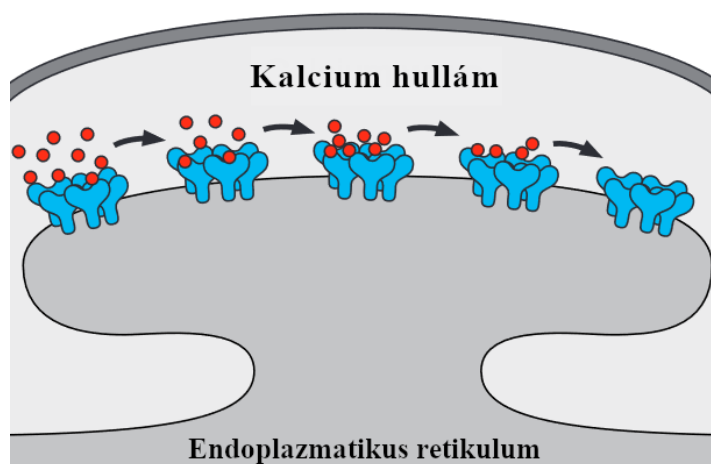
1. Bevezetés

1.1. Kalcium a szervezetben

A kalcium a szervezet fontos biogén eleme, mely a sejt létrejöttétől egészen a haláláig szerepet játszik mechanizmusában. Szerepe a sejt differenciálódásában, fejlődésében, aktivitásában és a sejthalál kialakításában is igen jelentős [1]. Hatását intracelluláris jelátvivő anyagként fejt ki, jelátviteli útjai pedig igen változatosak, ami lehetővé teszi széleskörű szabályozó feladatainak ellátását. Ezt a változatosságot a kalciumhoz kötődő folyamatok négy tulajdonsága együttesen határozza meg, melyek a Ca^{2+} származása, a válasz időtartama, a kalcium szint változásának amplitúdója és frekvenciája.

A Ca^{2+} két térségből juthat a citoplazmába, az extracelluláris térből éppúgy, mint az intracellulárisból. A sejten kívüli térből három féle csatorna engedi át a sejtmembránon, ezek a feszültségfüggő csatornák, amik főleg ideg- és izomsejtekben találhatók, a ligandfüggő csatornák és a raktárfüggő csatornák [1, 2]. Az intracelluláris tér kalcium raktára az endoplazmatikus retikulum (izomsejtekben a szarkoplazmatikus retikulum ennek megfelelője). Az innen történő ürülést biztosító két legfontosabb receptor az inozitol-1,4,5-trifoszfát (IP_3) és a kifejezetten Ca^{2+} hatására nyíló rianodin receptor [1, 2].

A válasz időtartama széles skálán mozoghat a milliszekundumos nagyságrendtől az akár egy órán keresztül tartó kalcium szint változásig, ami jelentősen befolyásolja, hogy milyen reakciók indulnak meg a sejtben. A gyorsan lezajló folyamatok lokalizált Ca^{2+} koncentráció növekedést tesznek lehetővé. Ezeket elemi jeleknek nevezzük, amik a kis kiterjedésük miatt specifikus reakciókat válthatnak ki a sejtben, például a szinaptikus vezikulák exocitózist [2]. Az ilyen elemi jelekből a kalcium indukálta kalcium felszabadulás által kialakulhatnak az egész sejtre kiterjedő és hosszan tartó globális jelek. Ennek alapja, hogy az elemi jel okozta lokálisan megnőtt Ca^{2+} koncentráció a kalcium diffúzióját idézi elő a citoplazmában. Az így elért közeli receptorok koordinált aktiválása újabb Ca^{2+} felszabadulásokat vált ki. Ennek a folyamatnak a része, hogy a felszabadult Ca^{2+} hatására az IP_3 receptorok érzékenyítődnek, így még könnyebben nyílnak. Az elemi jelek ezáltal kialakíthatnak egy, az egész sejten tovaterjedő hullámot, vagy akár oszcilláló koncentrációváltozást is (1. ábra) [2].



1. ábra: A kalcium indukálta kalcium felszabadulás. Az elemi jel által kiváltott Ca^{2+} ionok (piros) a citoplazmába jutva diffúzióval mozognak. Az endoplazmatikus retikulum receptorait elérve (kék) érzékenyítik azokat, így megkönnyítve a további Ca^{2+} felszabadulást. A folyamat lehetővé teszi a távolabbi receptorok kalcium felszabadítását is. A kép a [2]-es forrásból került átvételre, a rajta elvégzett változtatások során a feliratot magyarra fordítottam és az a) megjelölést töröltem róla.

A globális jelek esetében a koncentráció szint változása jellemezhető az amplitúdó nagyságával is, aminek egyes gének aktiválásában van fontos szerepe [1]. Általános, hogy nehéz ennek kis változásait észlelni, azonban kutatások alapján elmondható, hogy a sejt megfelelően képes az eltérések érzékelésére. Oszcilláló koncentrációváltozás esetén a frekvencia is meghatározó lehet. Növekedett frekvencia például az artériák simaizomsejtjeinek elernyedését okozza. [1].

1.2. Kalcium szint monitorozása

Az intracellularis kalcium szint monitorozása és annak megfelelő vizualizálása sokban hozzá tud járulni a sejten belül lezajló folyamatok feltérképezésében és megértésében. Ezt felismerve, az elmúlt évtizedekben sokan dolgoztak különféle indikátorok kidolgozásán, melyekkel a sejten belüli Ca^{2+} koncentráció megfelelően mérhető.

A kalcium indikátorok három csoportba sorolhatók, mely csoportok a biolumineszcens indikátorokat, a kémiai indikátorokat és a genetikailag kódolt kalcium indikátorokat (GKKI) foglalják magukban.

Legkorábban a biolumineszcens indikátorokat, mint az *Aequoria victoria* medúza fajban megtalálható aequorint, fedezték fel [3]. Az aequorin egy apoproteinből (apoequorinból) és a hozzá csatlakozó színhordozó csoportból (coelenterazin és az azáltal megkötött

molekuláris oxigén) épül fel. A molekulában megtalálható három kalcium kötő helyre Ca^{2+} bekötődése esetén a térszerkezet irreverzibilis konformációváltozása következik be, amit a coelenterazin coelenteramiddá történő oxidációja kísér, energia leadás mellett. Ez az energia egy foton formájában távozik, a folyamat során kibocsátott fény hullámhossza pedig 470 nm [4].

A biolumineszcens indikátorok használatához nem kell külső fénnel gerjeszteni a molekulát, így kiküszöbölve az autofluoreszcenciát és a sejtet érő káros külső sugárzást (például UV). A biolumineszcens fényre nem jellemző az úgynevezett fotoelhalványodás (angolul photobleaching) jelensége sem, melynek során a molekula fénykibocsátó tulajdonsága fokozatosan megszűnik [4]. Mivel a sejtben a biolumineszcens molekulán kívül nincs autofluoreszcencia, így az ilyen festékek jel-zaj aránya igen jó, hiszen az emittált fény kizárólag az indikátortól fog származni [3].

Az aequorin használatát körülményessé teszi azonban, hogy a molekula csupán egyetlen foton kibocsátására képes, ezenfelül a festéket külsőleg kell bejuttatni a sejtbe, például mikropipetta használatával [3, 4].

A második csoportot a kémiai indikátorok képviselik. Ezekben az indikátorokban egy kalcium fogó kelát és egy színhordozó csoport kovalens kötéssel kapcsolódik, így alkotva molekulát [4]. A biolumineszcens festékekkel ellentétben, a kémiai indikátorokat külső fény, legtöbbször UV, besugárzásával kell gerjeszteni. Az általuk kibocsátott fény hullámhossza az aequorinnál nagyobb, általában 505-520 nm [4]. Működésük azon alapul, hogy a kelát reverzibilis Ca^{2+} megkötése után a molekula térszerkezete változik, ami az általa emittált fény hullámhosszának megváltozását is okozza [4].

A kémiai indikátorok használata mellett és ellen is szólnak érvek. Előnyük, hogy általában jó a jel-zaj aránya a velük készült felvételeknek, illetve ma már rengetegféle kémiai indikátor létezik, melyek Ca^{2+} affinitása és fényspektruma széles határok között húzódik [4]. Ezekkel az indikátorokkal szinte minden típusú sejt festhető.

Hátrányos tulajdonságaik közé tartozik, hogy működésükhöz külső fénybesugárzás szükséges, mely legtöbbször UV besugárzást jelent. Ez okozhat a sejtben fototoxicitást, valamint autofluoreszcenciát is, ami a jel-zaj arányt csökkentheti [3, 4]. Az ilyen indikátorok nem használhatók a több napon keresztül elhúzódó mérésekhez sem, emellett itt is körülményes a festéket a sejtekbe juttatni. Erre ugyan születtek az évek során megoldások (mint az acetoxi-metil észter alkalmazása, amely formában a festék át tud jutni a sejtmembránon, a citoplazmába jutva pedig aspecifikus észteráz enzimek lehasítják az acetoxi-metilt róla), azonban még ez is bonyolítja használatukat [3].

A harmadik csoport tagjai a genetikailag kódolt kalcium indikátorok (GKKI). Két típusukat lehet megkülönböztetni, aszerint, hogy egy vagy kettő fluorofór (fluoreszcens protein) molekulát tartalmaznak. A két fluorofórt tartalmazó GKKI-k működési elve a Fröster-féle rezonáns energiaátadásra (röviden FRET) épül, amely egy nem radioaktív energiaátadás a donor és az akceptor fluorofór között [4]. A FRET létrejöttéhez kulcsfontosságú, hogy a két fluorofór egymástól megfelelő, legfeljebb 10 nm, távolságra legyen [4, 5]. A két proteint egy calmodulin (Ca^{2+} kötő fehérje) és egy M13 (calmodulin kötő peptid) kapcsolja össze.

Mikor a calmodulin megköt egy kalcium iont, a molekulában konformációváltozás megy végbe, melynek folyamán a két fluorofór közelebb kerül egymáshoz, így elérve a megfelelő távolságot. A donor molekulából kilépő fotonok ezáltal gerjeszteni tudják az akceptor molekulát, ami így szintén fényt fog emittálni. Fontos, hogy a donor molekulát ehhez a folyamat során külső fénybesugárzással gerjeszteni kell, továbbá a két fluorofórt úgy kell megválasztani, hogy a donor által kibocsátott fény gerjeszteni tudja az akceptor molekulát [3]. A kalcium szint mennyiségére az akceptor és a donor fluorofór által emittált fény intenzitásának arányából tudunk következtetni [5].

A calmodulin nem szelektív kötésének kiküszöbölésére születtek változtatások az alap felépítéshez képest, például a calmodulint gyakran cserélik ki tropionin C molekulára [5].

Egy fluorofórt tartalmazó GKKI-k esetében a calmodulin és az M13 a fluoreszcens protein két oldalán helyezkednek el. Ebben az esetben is a calmodulin Ca^{2+} megkötése okozza a térszerkezetváltozást, azonban itt csak az emittált fény intenzitása növekszik, a hullámhossz változatlan marad [5]. Éppen ezért ezek a GKKI-k nem olyan megbízhatóak, mint a két fluorofóros változatok.

A GKKI-kat azért hozták létre, hogy a sejteket specifikusan lehessen jelölni, ami a kémiai és biolumineszcens festékek esetében nem mindig lehetséges. A GKKI elnevezés is utal arra, hogy ezeket az indikátorokat a sejt maga írja át a DNS-éből, így specifikusan lehet olyan sejteket előállítani, amiket jelölni szeretnénk. A gén bejuttatása történhet virális vektorok segítségével [4]. Előnyei ezenfelül, hogy hosszabb ideig lehet velük jelölni a sejteket [5] és a kalcium megkötése reverzibilis úton történik [3].

Negatívumok itt is felmerülnek. Pontosan kell ismerni a kódot, amiről az indikátor átíródik, emellett az aktív génállományt is, mivel az indikátor génje csak az aktív szakaszba épülve fog átíródni. Ezután az indikátorral felszerelt sejteket vagy akár állattörzseket a kutató(k)nak kell tenyészteni. Ennél az indikátor típusnál is szükség van külső besugárzásra, éppúgy, mint a kémiai indikátorok esetében [3].

Az eddigiekből látszik, hogy ezen indikátorok a citoplazmában lévő szabad Ca^{2+} kötésére képesek, mely ionok egyensúlyban vannak a kötött formákkal [4]. Mivel az indikátorok is megkötik a kalciumot, így valójában felborítják ezt az egyensúlyt, ezáltal a belső dinamikákba is beleszólhatnak [4]. A mérések során ezt mindenképp figyelembe kell venni.

A kutatás tervezésekor ezért nagy szerepe van a megfelelő indikátor kiválasztásának. Érdekes lehetőség szerint alacsony affinitású indikátort választani, mert az kevésbé fog beleszólni a sejten belüli egyensúlyba. Az alacsony affinitású festékkel festett sejtről azonban csak a megfelelő érzékenységgű műszer tud jó minőségű képet alkotni, így zajos kép esetében a magas affinitású indikátor jobb választás lehet [4].

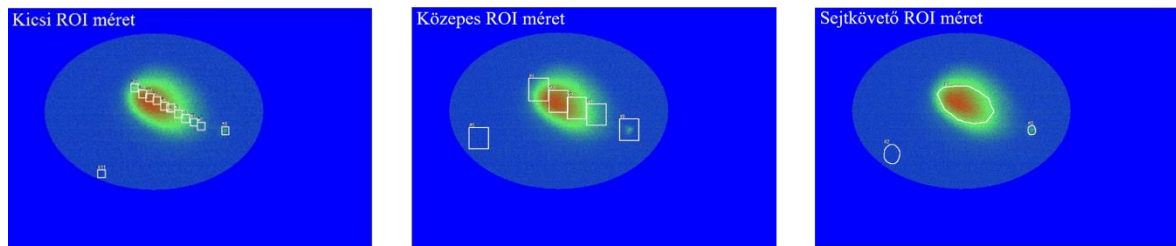
1.3. A felvett adatok elemzése

Az előző alfejezetben tárgyalt indikátorok segítségével a sejtekben lezajló folyamatokat videó formájában rögzíteni lehet. Ezek a videók aztán további elemzésre számítógépes szoftverekbe táplálhatók. Ilyen szoftverekből nincs sok, közülük pedig a legtöbb ingyenesen nem is érhető el. Gyakori probléma, hogy a fejlesztő cégek vagy megszűntek, vagy már nem ölnek annyi energiát a program folyamatos fejlesztésébe, így ezek a szoftverek egyre elavultabbak, esetlegesen már nem is kompatibilisek a legújabb operációs rendszerekkel. Az egyik legelterjedtebb ilyen szoftver, az Imaging Workbench, is hasonló helyzetbe került. Utolsó nagy frissítésére 2007-ben került sor [6], majd a fejlesztő cég megszűnt. Fejlesztése ugyan átkerült egy másik céghez, az INDEC BioSystems-hez, viszont azóta csak egy kisebb frissítése lett lefejlesztve [7].

Az elemzési folyamat egyik legfontosabb lépése a felvételen vizsgálandó területek kijelölése. Az angol szakirodalom a „region of interest” kifejezést használja, melyből a ROI rövidítés adódik. A továbbiakban én is ezt a rövidítést fogom alkalmazni.

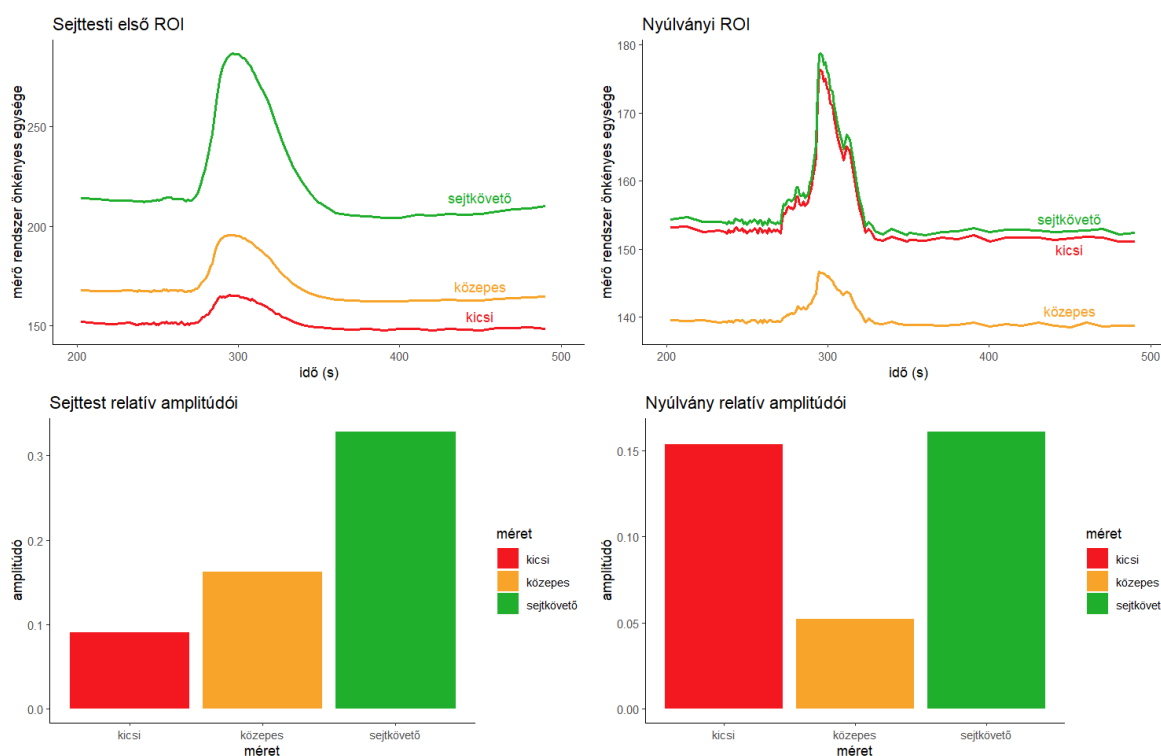
A programokban közös, hogy az egyedi sejtek vizsgálatánál a ROI-k felhelyezése kézzel történik, amely több hátrányát is okozza ezeknek a szoftvereknek. A felhelyezésben jelentős szerepet kap az emberi hiba, hiszen szinte lehetetlen kétszer ugyanarra a pixelhalmazra helyezni a ROI-t, így csökken a kísérlet megismételhetősége. Emellett az egymás mellé felrajzolt ROI-k méretei is eltéréseket mutathatnak, hiába törekszik a felhelyező személy az azonosságra. A kutatók körében nincs arra egységes megállapodás, hogy mekkorára és milyen alakúra kell az egyes ROI-kat megrajzolni. A 2. ábra bemutat néhány lehetséges felhelyezést. Jól látszik, hogy a méret a felhelyezhető ROI-k számát is jelentősen

befolyásolja. Egy a sejten tovaterjedő hullám vizsgálata során ez okozhat mérési különbségeket.



2. ábra: A ROI-k felhelyezésének néhány lehetséges formája a nyúlvánnyal rendelkező Deiters sejtre. A bal oldali kép a kicsi (nyúlvány nagyságú) ROI méretet, a középső a közepes méretet ábrázolja, a jobb oldali képen pedig egy nagy, sejtkövető ROI, és egy kisebb, nyúlványt lefedő ROI van. Mindhárom képen látszik a háttérben is egy ROI, ez az intenzitásértékek korrekciójához szükséges.

A 3. ábra a különböző méretekből adódó eltéréseket szemlélteti a már kiszámolt intenzitásgörbék bemutatásával.



3. ábra: A különböző méretű ROI-k által okozott különbség a Deiters sejtek intenzitásgörbéin. A sejtet a kísérlet során 100 μ M ATP-vel ingereltük, ami purinerg receptorokon keresztül az intracelluláris Ca^{2+} szint emelkedését okozza. Az intenzitások a sejttest esetében csak az első ROI területén vannak ábrázolva. A felső képek a vizsgálat alatt lezajlott intenzitásváltozást mutatják, az alsó képek pedig a relatív amplitúdókat hasonlítják össze. A bal oldali kép mindig a sejttesten lévő első ROI adatait mutatja, a jobb oldali pedig mindig a nyúlványét.

Jól látszik, hogy a ROI mérete és az ebből adódó helyzete nagy különbségeket okoz az érzékelt intenzitás nagyságában és a relatív amplitúdóban is. A felvétel során a legintenzívebb válasz a sejttest közepén volt látható, ezért a sejttest szélére került kicsi ROI területén volt a legkisebb a fluoreszcencia változás. A közepes ROI esetében a vizsgált terület már jobban benyúlt a sejt közepe felé, ezért itt nagyobb mértékű intenzitást mértünk. A legnagyobb relatív amplitúdó a sejtkövető ROI esetében adódott, hiszen itt a sejt közepe is mérésre került, illetve ebben az esetben az alapintenzitás is magasabb volt, hiszen itt a ROI arányaihoz képest kevesebb volt a ROI-ba belógó háttér területe.

A nyúlvány vizsgálatában a kicsi és a sejt követő ROI intenzitás görbéje szinte megegyező volt, mert a kicsi méret a nyúlvány mérete szerint lett meghatározva.

Ezek alapján jól látszik, hogy a ROI mérete és alakja nagy befolyásoló hatással bír, különösen, ha kis kompartmenteket is vizsgálni szeretnénk. Ilyen kompartment lehet például képeken látott Deiters sejt nyúlványa, egy fotoreceptor, vagy egy idegsejt, illetve gliasejt nyúlványa is. Fontos tehát, hogy egy egységes rendszer legyen a vizsgálandó területek kijelölésére.

A ROI-k felhelyezése mellett van egy másik szubjektív tényező is. A felvételen gyakori, hogy a sejt elmozdul, azonban sok szoftver nem tudja lekövetni ezt. Az ilyen programokban az elemző személy kézzel állítja a felvétel során a ROI-k helyzetét úgy, hogy az általa észlelt elmozdulást korrigálja aszerint, hogy a ROI-k ismét a vizsgálandó helyre kerüljenek. A szubjektív oldalát az adja, hogy nem mindenki tudja ugyanolyan érzékenyen észlelni az elmozdulást, ezenfelül a ROI-k is csak bizonyos távolságoként mozgathatók, nem lehet őket finoman hangolni.

Ehhez hozzájön az is, hogy a program nem nyílt forráskódot használ, így az elemző nem tudja lekérdezni a számítási algoritmusokat. A 3. ábrán is olvasható, hogy a program önkényesen skálázta az intenzitás értékeket, a különböző programok pedig eltérő skálákat használnak.

Mások is felismerték a fent említett problémákat, így született néhány olyan program, melynek indíttatása hasonló a későbbiekben bemutatásra kerülő algoritmusával. Ilyen például az EzCalcium [8], a SIMA [9], a Mesmerize [10], a CaImAn [11] és a Suite2p [12]. Ezek mind vagy MATLAB vagy Python nyelven lettek kódolva, R-ben még senki nem írt hasonló programot. Emellett a felsoroltak közül több is felhasználja valamelyik másiknak az algoritmusát egyes lépésekhez, nem csak saját kódolású mechanizmusokat [8, 10]. A jelenleg létező programok közül sok idegsejtek vizsgálatára lett fejlesztve, így ezeknél a cél az, hogy minél több sejtet el tudjon különíteni, nem egy sejt elemzésére találták ki őket.

2. Célkitűzések

A jelenleg rendelkezésre álló számítógépes szoftverek használata sok negatívummal jár. Ingyenesen nem érhető el, cserébe elavultak. Fejlesztésük jobb esetben is kimerül annyiban, hogy a jelenlegi operációs rendszerekkel kompatibilissé teszik őket. Az elemzési munka sok szubjektív tényezőt foglal magában, ami csökkenti a kísérlet megismételhetőségét, és befolyásolja az eredményeket. A program kódjához az elemző nem fér hozzá, ezért nehezen testreszabható a vizsgálat, a számítási folyamatok és algoritmusok pedig nem kérdezhetők le. Nem tudni tehát, hogy mi alapján skálázza az adatokat, továbbá szoftverek között is eltérés lehet a skálázásban és az eredmények számítási módszereiben. Sok szoftverben az sincs megoldva, hogy az elemzés automatikus legyen, ezért a folyamat idő- és emberigényes, ami kritikus lehet egy kutató életében.

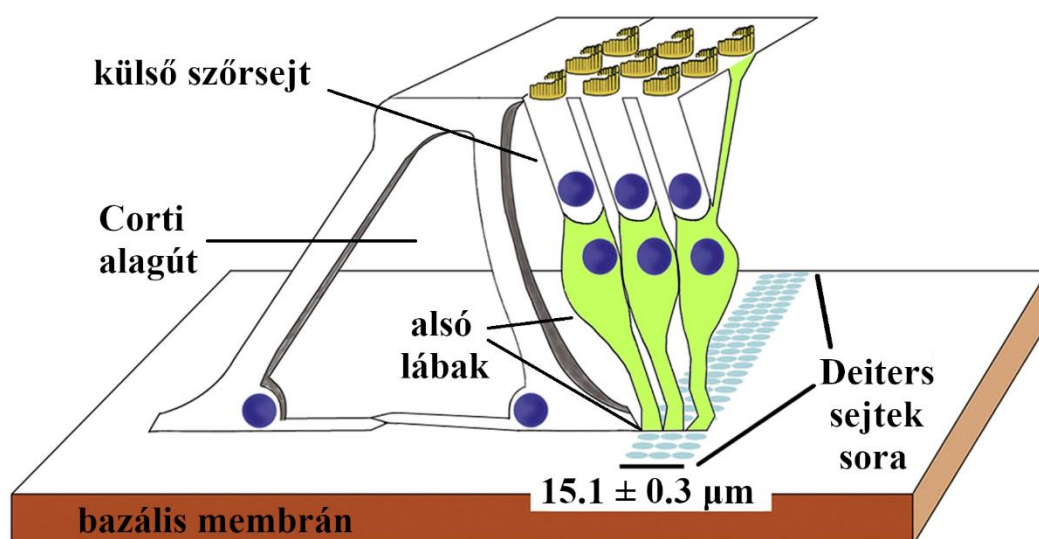
A felsorolt hátrányok alapján született meg a terv egy olyan algoritmusra, ami ezeket kiküszöböli. Cél volt az ingyenes elérhetőség és a nyílt forráskód biztosítása, amely talán a két legfontosabb aspektusa egy ilyen programnak. Ezek biztosítják, hogy könnyen elérhető és bárki által testreszabható legyen, ami a széleskörű használatot teszi lehetővé. Az objektivitást és az automatizálást is szem előtt tartva igyekeztünk a lehető legkevesebb interakciót igénylő programot írni, aminek futási ideje is optimális.

A munkához az R programozás nyelvet használtuk, mert célunk volt a biológia területén dolgozó legtöbb kutató elérése, akik közül sokan használják ezt a nyelvet. Szempont volt, hogy minél kevesebb R csomag kelljen a használatához, hiszen azokat bárki módosíthatja, ezáltal változás állhat be a program által használt kódok működésében is.

3. Anyag és módszer

3.1. Deiters sejtek

A Deiters sejtek a belső fülben található Corti-szerv egyik típusú sejtjei. Fő feladatuk a külső szőrsejtek alulról történő támasztása [13]. A sejtek rendelkeznek egy nyúlvánnyal is, ami a szőrsejtek csúcsi részéig felér, és akár két-három sejttel arrébb lévő szőrsejthez is elnyúlhat [14]. Általános felépítésüket a 4. ábra illusztrálja.



4. ábra: A Deiters sejtek általános felépítése és elhelyezkedése a Corti-szerven belül. Az ábrán látszik, hogy a Deiters sejtek (zöld) a külső szőrsejtekhez hasonlóan több sorban helyezkednek el. Ezen sorok együttes vastagsága a Deiters sejtek lábainál $15.1 \pm 0.3 \mu\text{m}$. A kép az eredeti formájához képest változtatva lett, a feliratok magyarra fordítása, illetve kiegészítése és a Deiters sejtek színezése révén [13].

A program megírását ezen sejtek vizsgálata ösztönözte. A Deiters sejtekkel kapcsolatos kutatásainkban a sejtben lezajló Ca^{2+} szint változásokról felvétel készül. Ehhez fluoreszcens (oregon green BAPTA-1, Sigma Aldrich) indikátorral megfestett sejteket használunk.

3.2. Az R programozási nyelv

Az algoritmus az R programozási nyelven íródott. Az R egy nyílt forráskódú nyelv, így bárki hozzáférhet a kódhoz, szabadon változtathat rajta, hogy a lehető legjobban saját igényeire formálja. A működése az S statisztikai nyelven alapul, azonban a megalkotói nem csupán statisztikai nyelvként szeretnek az R-re gondolni [15]. A nyelv elérhető Windows, macOS és Linux operációs rendszereken is.

A biológia területén az egyik legelterjedtebb nyelv, mert statisztikai elemzéseket remekül lehet benne végezni. Ez köszönhető a rendkívül sok beépített statisztikai módszernek, illetve annak, hogy az eredményeket látványos grafikonokon is szemléltetni lehet, rögtön az R-en belül [15]. Ebben sok nyelvtől különbözik, mert csak kevés másiknak van hasonlóan funkcionális grafikus panelje. Másik erőssége a könnyen használható segítségkérő funkciója. A `help()` parancsot alkalmazva egy részletes leírást kapunk az éppen keresett kérdéssel kapcsolatban [15].

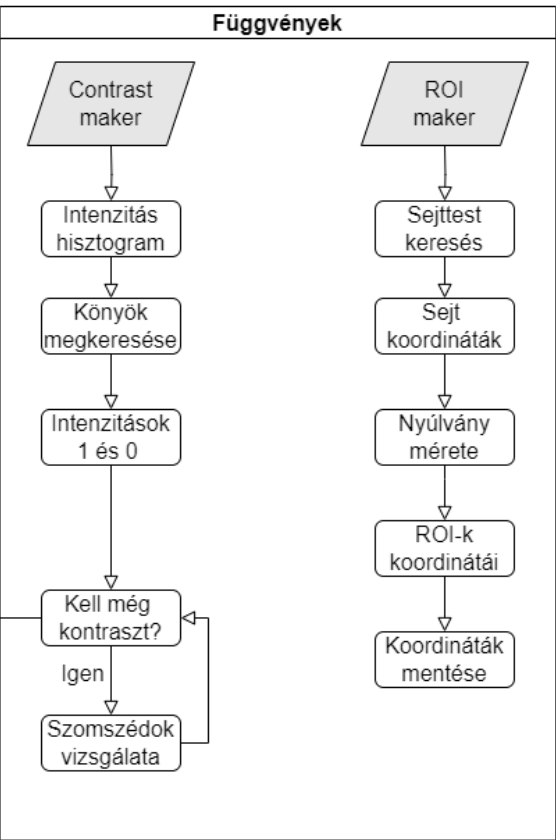
Az R úgynevezett csomagokat használ, amik az alap funkciókat bővítik ki. Az R telepítésekor alából 8 ilyen csomag töltődik le, amik a legegyszerűbb programozási feladatok elvégzését teszik lehetővé [15]. Az alap csomagokban található kódok azonban sokszor nem elegendőek speciálisabb programok megírásához. Az ilyenkor kellő kódokat a különféle csomagokban lehet megtalálni, melyeket a felhasználó szükség szerint maga tölt le. A nyílt forráskód miatt lehetőség van saját csomagok megalkotására is, ami magasfokú testreszabhatósággal jár együtt [15].

A felület, amit a kód megírásához használtunk, az RStudio nevű platform, ami az R-en kívül több nyelvet is kezelni tud. Az RStudio igen felhasználóbarát, egy felületen egyesíti a programozás során használt rétegeket, mint a környezetet, ahova a különböző változók is kerülnek, a munkakönyvtárban megtalálható fájlokat, a konzolt és magát a kódoló ablakot is [16].

3.3. A kód működése

Az általunk használt R csomagok: `tiff`, `grid`, `scales`, és a `quantmod`. Ahogy az első csomag is utal rá, az beolvasandó fájlnak TIFF multipage formátumúnak kell lennie. A TIFF beolvasása után minden kép egy külön réteggént lesz elmentve egy listában. A lista elemei ezáltal olyan mátrixok lesznek, amik a kép pixeleinek intenzitásait tárolják magukban. Ezután a függvények lefuttatása a következő lépés, amelyeket később a fő algoritmus majd felhasznál. A függvényekben lévő kódokat azért kell külön kezelni a fő programtól, mert ezek több helyen is meghívódnak, így sokkal letisztultabb a program, ha a függvények kívülről vannak behívva. Az általunk írt és használt függvények: `kneedle`, `contrast maker`, `rotation`, `ROI maker`, `ROI calculator`, és `inflect`. Ezek részletes működése a következő alpontokban kifejtésre kerül. A függvények beolvasása után a fő algoritmus futtatása következik. A program működésének összefoglalását az 5. ábra mutatja be. A program végén az adatok data frame formájában kerülnek mentésre, mert így könnyen kezelhető az

5. ábra: A program működését összefoglaló folyamatábra. A két legtöbb lépést tartalmazó függvény (Contrast maker és ROI maker) külön jelölve van a lépések feltüntetésével.



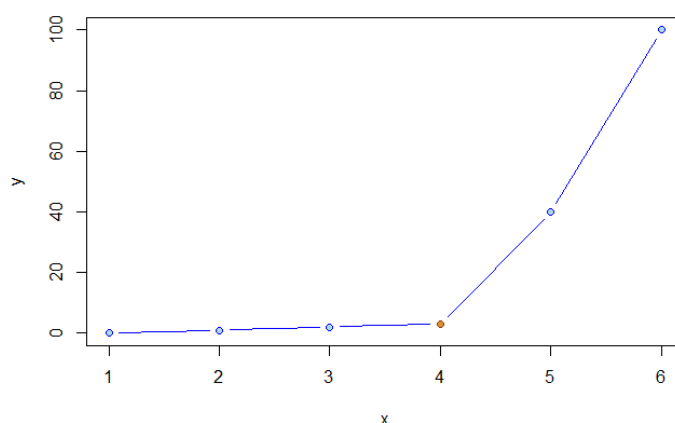
3.3.1. Start

A program indítása után rögtön egy logikai vizsgálat következik, ami megnézi, hogy az első rétegnél jár-e az algoritmus. A futási idő optimalizálása miatt a legtöbb lépés csak az első réteg esetében zajlik le. Az itt kiszámolt értékek jórészt használhatók a többi rétegre is, így nem kell minden új rétegnél a számolással időt veszíteni. Az első réteg esetén a program tovább ugrik a contrast maker nevű függvényre.

3.3.2. Kneedle

A folyamatábrán ezt a függvényt reprezentálja a „Könyök megkeresése” nevű alakzat a contrast maker függvényen belül. A programban ez az egyetlen mechanizmus, amit nem mi programoztunk le, hanem egy már meglévő, Emmet Tam által elérhetővé tett algoritmust használtunk fel [17].

Az algoritmus a felhasználandó adatokat vektor formájában kéri be. Két vektort kell betáplálni, amik együttesen koordinátákat adnak ki (tehát kell egy vektor, ami az x-ket és egy, ami az y-kat tartalmazza) [17]. Ezeket az algoritmus az x-k szerint sorba rendezi, és az így kapott görbén megkeresi a „könyököt” (6. ábra). Ehhez az első és a második deriváltat felhasználva azt fogja végül könyöknek érzékelni, ahol a deriváltak segítségével számolt görbület maximális [18]. A könyök koordinátáját aztán a függvény kimenetként visszaadja, és lehet vele tovább dolgozni [17].

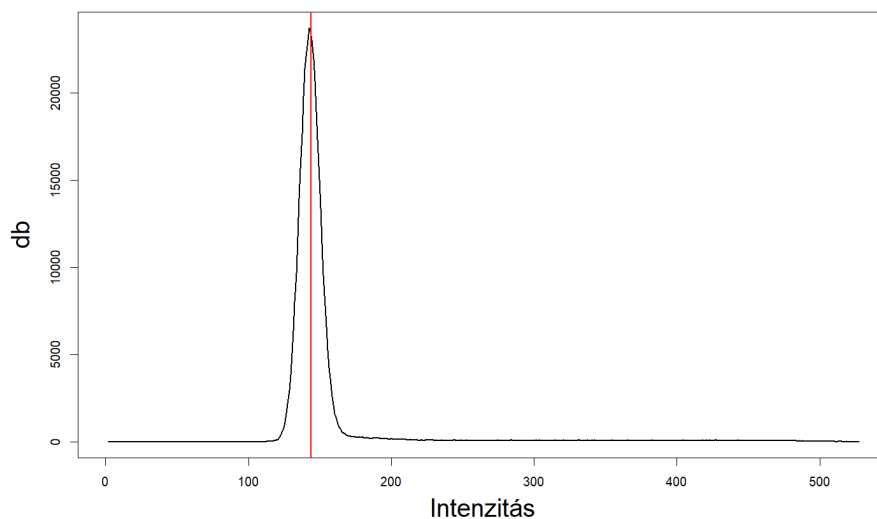


6. ábra: A könyök szemléltetése. Ezen az ábrán a könyököt a piros pont jelöli. Az ábra az algoritmus készítőjétől származik, a piros pont utólag lett átszínezve [17].

3.3.3. Contrast maker

A függvény feladata az intenzitásokat kicserélni úgy, hogy a mátrixban csak 0 és 1 értékek szerepeljenek. A sejttest így optimális esetben csupa 1-es, a háttér pedig csupa 0-ás értéket kap.

Ehhez a képet mátrix formájában beolvassa, a pixelek intenzitásértékeiből először egy hisztogramot készítünk (7. ábra), az így kapott görbe pontjainak koordinátáit pedig betápláljuk a kneedle függvénybe. A hisztogram egy unimodális görbét eredményez, aminek a maximuma a háttérintenzitásoknál van, mert a háttér (szürke) intenzitásértékei képviselik az adatok legnagyobb részét. A sejttest értékeit a csúcs utáni ellaposodó szakasz reprezentálja. Ahhoz, hogy a kneedle függvény megfelelően kezelje ezt a görbét, a felvételek (tárhely kímélő) körülvágását, ahol az intenzitás 0, nem vettük bele a hisztogramot alkotó adatokba. Emellett a kneedle függvénynek csak az intenzitások mediánjánál nagyobb intenzitásértékek adódnak át, ezzel biztosítva, hogy a görbén csak egyetlen könyök szerepeljen.



7. ábra: Az intenzitásértékekből készített hisztogram. A piros vonal mutatja a medián értékét a 0 intenzitásértékeket nem nézve. A kneedle algoritmus a piros vonaltól jobbra eső adatokat kapja meg, ezzel biztosítva, hogy csak egy könyök legyen az adatokban.

A kneedle függvény a megkapott adatokból a fentiek szerint visszaadja a könyök koordinátáit. Ez a koordináta adja majd a kontrasztozáshoz azt az intenzitásértéket, amely alatt a mátrixban tárolt intenzitások 0 értéket kapnak, a könyök értékénél nagyobb intenzitások pedig mind 1-es értéké válnak. Az így kapott mátrixot a függvény kirajzolja,

és megkérdezi, hogy szükséges-e még további kontrasztosítás. Nemleges válasz esetén elmenti a mátrixot és kilép a függvényből. Igen választ kapva azonban további értékek nullázódnak ki, ha nincs legalább három olyan szomszédos cellájuk, melyben 1-es érték szerepel. A program az így kapott új mátrix kirajzolása mellett ismét rákérdez a további kontrasztosításra. A szomszédok vizsgálatát folyamatosan igent válaszolva egymás után többször is el lehet végezni. Tovább lépni nemleges válasszal lehet. A lépések eredményeit a 8. ábra szemlélteti.



8. ábra: A kontrasztosítás lépései. A bal oldalon az eredeti kép látható, a középső kép a 0-1-es közötti küszöbérték meghatározása után készült, a jobb oldali kép pedig az első további kontrasztosítás eredményét mutatja. A program a további kontrasztosítások számát késsel ráírja a képre. Látható, hogy az első kontrasztosítás is igen jó eredményt ad, azonban jobban megnézve látszik a sejtest mellett a dudornál egy kis szóródás. A további kontrasztosítás ezt csökkentette.

3.3.4. A forgatási szög számolása

A kontrasztosítás után kapott mátrix értékeihez regressziós egyenest illeszt a program. A kontrasztosítás ezért szükséges előfeltétele a forgatási szög keresésének, hiszen a contrast maker kimeneteként kapott adatformára lehet megfelelő regressziós egyenest illeszteni. Az egyenes meredekségéből megkapjuk a forgatáshoz kellő szöget. A sejtet a további lépésekhez nagyjából vízszintes helyzetbe kell fordítani, mert ezáltal küszöbölhető ki az átfedés a nyúlvány intenzitásait tartalmazó oszlopok és a sejtest értékeit tartalmazók között. Ehhez a regressziós egyenes elegendően pontosnak bizonyult.

3.3.5. Forgatás

A forgatást a program forgatási mátrix és bilineáris interpoláció segítségével végzi. A forgatási mátrix egy olyan négyzetes mátrix, ami ortogonális és a determinánsa 1 [19].

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Egy ilyen mátrix egy kétdimenziós oszlopvektort tud forgatni, ez az oszlopvektor az x és y koordinátákat tartalmazza.

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$

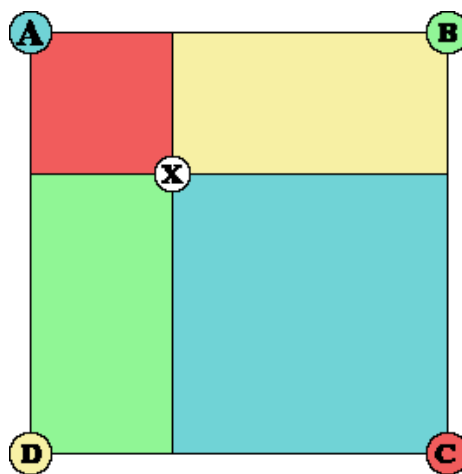
Ilyen formában a mátrix szorzás a nullvektorral a nullvektort fogja adni, tehát ez a forgatás az origó körüli forgatást jelenti. Így a koordinátatengelyeket a mátrix közepére kell rendezni. A koordináták a mátrixból úgy adódnak, hogy a mátrix dimenzióit elosztjuk kettővel (mivel az origó középre került), így az x tengely a $[-\frac{\text{oszlopok száma}}{2}; \frac{\text{oszlopok száma}}{2}]$ intervallum lesz, az y tengely pedig ugyanígy a sorok számával. A mátrixhoz hozzáfűződik egy csupa 0 értéket tartalmazó sor vagy oszlop, ha az adott dimenzió nem páros. Mivel a mátrix sorainak és oszlopainak száma lett egy-egy koordináta, így az eddig cellákban tárolt intenzitásérték most két koordináta közé esik félúton, így ezek x és y koordinátái valójában a síknegyedről függően +0,5 vagy -0,5 értékkel változnak. (A 2. és 3. síknegyedben +0,5-tel, az 1. és 4. síknegyedben pedig -0,5-tel.) Első lépésben a mátrix négy sarkában lévő érték koordinátáit forgatja a program a mátrix szorzás alapján.

A 0,5-tel eltolt koordinátákkal forgatva az új koordináták nem egész számok lesznek, viszont a mátrixban a sorok és oszlopok csak egész számokat vehetnek fel. Ahhoz, hogy az intenzitásértéket a megfelelő sorral és oszloppal lehessen majd hivatkozni, a program az intenzitásértékek új koordinátáinak a felső vagy alsó egészrészét veszi, attól függően, hogy melyik síknegyedbe esnek. A négy új koordináta által meghatározható egy új, csupa 0 értékkel feltöltött mátrix, aminek dimenzióit a négy forgatott sarok adja meg a 9. ábra szerint.



9. ábra: Az eredeti mátrix forgatva, és az általa meghatározható új mátrix. Az új mátrixot a forgatott mátrix négy sarka köré hozza létre a program.

A következő lépésben a program végigmegy minden új koordinátán (amit ugyanúgy alkot meg, mint a folyamat elején az eredeti mátrix esetében), és a forgatási mátrix segítségével, most $-\theta$ szöggel, visszaforgatja őket. Eközben megnézi, hogy visszaforgatva, beleesik-e az eredeti mátrix dimenzióiba, és ha nem, akkor meghagyja 0-nak az adott koordinátához tartozó értéket. Abban az esetben, ha beleesik, bilineáris interpoláció segítségével kiszámolja, hogy milyen intenzitásérték tartozik az adott új koordinátához. Azzal, hogy az új mátrix létrehozásakor az alsó és felső egészrészeket veszi a program, a visszaforgatásnál nem ugyanazt a koordinátát kapjuk, mint a forgatás előtti eredeti mátrixban. Így az új koordináta az eredeti mátrix négy ismert koordinátája közé fog esni (10. ábra).



10. ábra: Az új koordinátához tartozó ismeretlen érték visszaforgatva az eredeti mátrixban: X , és a négy ismert koordinátájú intenzitásérték: A , B , C , D .

Az új koordinátához tartozó intenzitásértéket a bilineáris interpoláció révén az alábbiak szerint kaphatjuk [20]:

$$X = (A \times \text{kék terület nagysága}) + (B \times \text{zöld terület nagysága}) \\ + (C \times \text{piros terület nagysága}) + (D \times \text{sárga terület nagysága})$$

A betűk intenzitásértékeit X -en kívül ismerjük, illetve a koordinátáját mindnek tudjuk, így a négyzetek területe könnyen kiszámolható, és ezáltal X értéke is. A függvény kimeneteként megkapjuk az új mátrixot, benne az új koordinátákhoz tartozó kiszámolt intenzitásértékekkel.

3.3.6. Nyúlvány helyzetének vizsgálata

A meredekséghez tartozó szöggel forgatva a Deiters sejt két irányban állhat az eredeti helyzetétől függően. A nyúlvány a sejttesttől vagy jobbra vagy balra kerülhet. Ezt a feltételt

az inflect függvény vizsgálja. A program szempontjából a nyúlványnak a jobb oldalra kell esnie a sejttestől, így, ha ez nem teljesül, a program 180° -t forgat a sejten. Ennek a folyamatnak a végén a megfelelő szög külön mentésre kerül, amit majd a következő rétegektől számolás nélkül rögtön lehet alkalmazni a forgatáshoz.

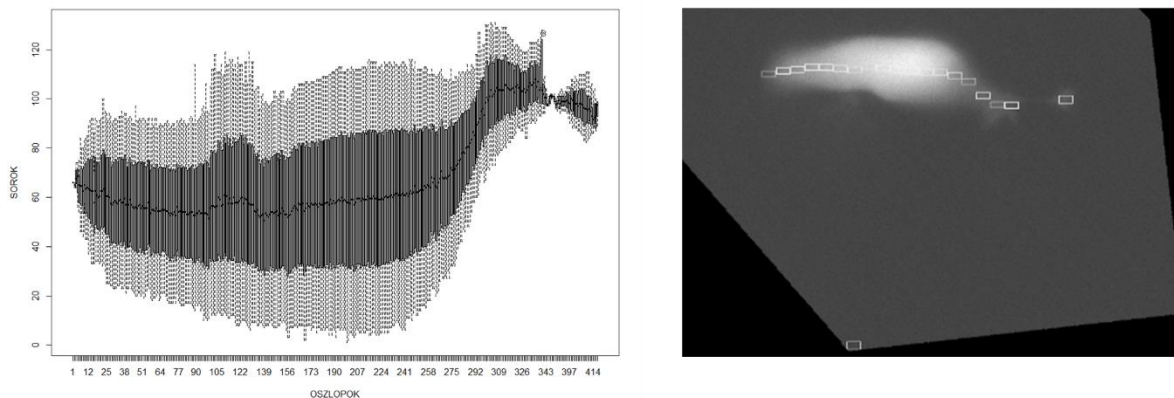
3.3.7. ROI maker

A ROI maker függvény felel a ROI-k helyzetének meghatározásáért. Ez a függvény sem hívódik meg az első után minden rétegre, csak ha a program egy későbbi szakaszban elcsúszást érzékelt.

Első lépésként megkeresi a kontrasztzott és már beforgatott mátrixon a sejttestet jelölő 1-es értékeket. Ezeket aztán kimentí egy data frame-be úgy, hogy minden 1-es érték mellé kimentí a hozzá tartozó sor- és oszlopszámot, amivel hivatkozni lehet rá az eredeti mátrixban.

A függvény ezután kirajzol egy újabb görbét, ami a kontrasztzott és beforgatott mátrix oszlopaiban lévő 1-esek számát ábrázolja. Az ábrán megjelenik három vonal is aszerint, hogy hány darab 1-est tartalmaz az az oszlop, amit a program már a sejt részének érzékel. Ezáltal tudjuk a ROI-k nagyságát meghatározni, amit érdemes a nyúlvány méretéhez igazítani.

A ROI-k helyzetét a program úgy határozza meg, hogy veszi az egyes oszlopokat, megnézi, hogy melyikben van a görbén beállított értéknél több. Ez kell ahhoz, hogy biztosan a sejt részeire kerüljenek a ROI-k, a nyúlvány és a sejttest közé pedig ne kerüljön. Ezt követően megnézi, hogy ezekben az oszlopokban milyen sorszámú sorokban van 1-es, és ezek mediánját veszi. Eköré a medián köré teszi fel a ROI-kat úgy, hogy a magassága az interkvartilis terjedelem, szélessége pedig a nyúlvány szélességével legyen egyenlő. Emellett a viszonyítási pont miatt a háttérre is kerül ROI. A pontos elemzéshez fontos, hogy minden ROI megegyező méretű legyen. A ROI-k száma attól függ, hogy mennyi fér rá a sejtre a nyúlvány méretű ROI-ból. A folyamat végén a függvény elmenti a ROI-k helyzeteit az eredeti képen (11. ábra).



11. ábra: A boxplot a mediánokat szemléltetve (bal) és a felpakolt ROI-k az eredeti forgatott képen (jobb). Látszik, hogy a mediánok szépen a sejt közepén húzódnak, illetve, hogy a nyúlvány és a sejttest között nincs felhelyezett ROI.

3.3.8. ROI calculator

Ez a függvény a folyamatábrán a „ROI-k átlag intenzitása” nevű alakzatként szerepel. A függvény feladata csupán annyi, hogy a ROI maker által kiszámolt és felpakolt ROI-kon belüli intenzitásértékeket és az azokból számolt átlagintenzitást elmenti, hogy később a további elemzéshez felhasználhatóak legyenek.

3.3.9. Elcsúszásvizsgálat

A programnak ez a része felelős azért, hogy a ROI-k mindig a megfelelő helyen maradjanak. A kísérletek során gyakran előfordul, hogy a sejt az eredeti helyéhez képest elmozdul, így kikerülhet a ROI-k által lefedett területről. Éppen ezért fontos, hogy az elcsúszás korrigálva legyen, ha szükséges.

A vizsgálat csak minden ötödik rétegnél történik meg, hogy a futási idő a lehető legoptimálisabb legyen. Wilcoxon-féle előjeles rangpróbát végezve nézi, hogy a ROI-k intenzitásértékeinek mediánjai mennyire térnek el a háttérétől. Szignifikáns különbség esetén nem veszi elcsúszásnak, azonban, ha a ROI-k negyedénél, vagy a nyúlványnál nem talál szignifikáns különbséget, újraszámolja a ROI-k helyzetét. Ilyenkor a program visszaugrik ROI maker függvényhez, és újra végigmegy a lépéseken.

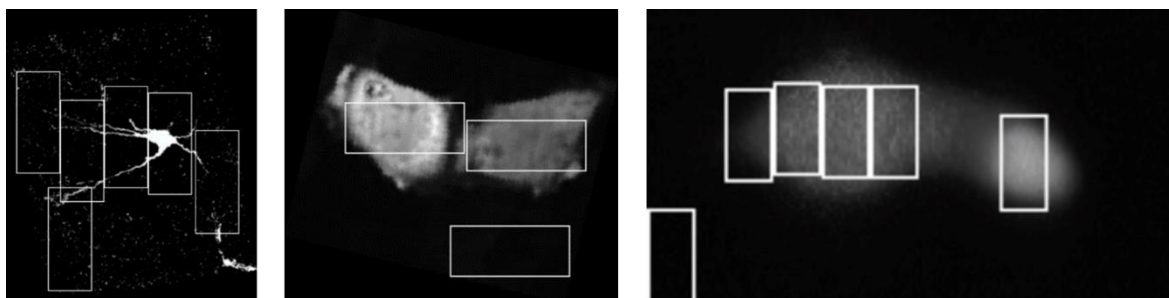
3.3.10. Mentés

A program végül egy data frame formájában elmenti a rétegszámot, a hozzá tartozó időt, a nyúlványon, a sejttesten és a háttéren lévő ROI-kon belüli átlagintenzitásokat, illetve, hogy azon a rétegen történt-e az elcsúszás miatt újraszámolás.

4. Eredmények

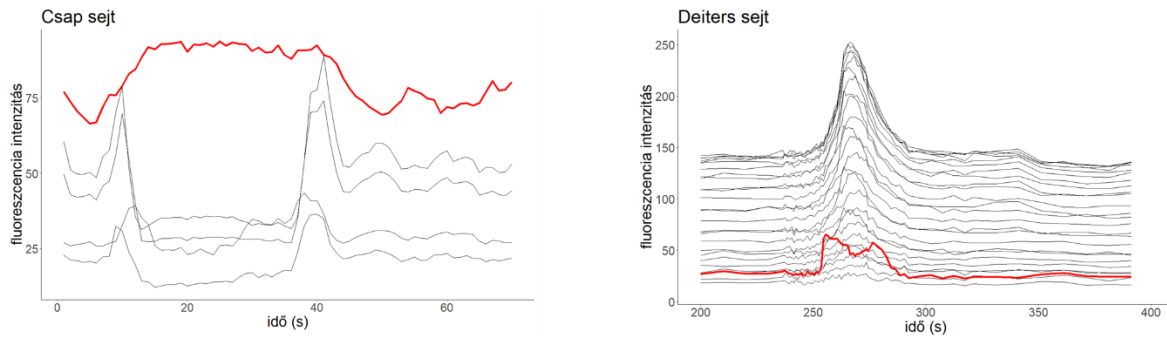
Az algoritmus sikeresen felállt, megfelelően lehet vele az intracelluláris Ca^{2+} jeleket követni mind a nyúlványon, mind a sejttesten belül. A ROI-k felhelyezése automatizált és objektív módszert használ a meghatározásukhoz.

A program a kódolás során főleg Deiters sejtekre lett optimalizálva, azonban más sejtekről készült felvételekkel is teszteltük az algoritmust, és ezekre is sikeresen elvégezte az elemzést. A sejtek, amivel tesztelve lett a program: szalamandra retinális csapsejt [21], patkány amygdalájának interneuronja [22] és HeLa sejtek [23]. A program mindhárom sejtre sikeresen felhelyezte a ROI-kat, illetve képes a nyúlványokat is megfelelően kezelni. A csapsejt esetében az is működött, hogy a nyúlvány és a sejttest közé nem tett fel ROI-t. A 12. ábra a tesztelés eredményét mutatja be.



12. ábra: Más sejtekkel lefuttatott tesztelés eredménye. Bal oldalon az interneuron [22], középen a HeLa sejtek [23], a jobb oldalon pedig a csapsejt [21] látható. A csapsejt esetében az is megvalósult, hogy a nyúlványt külön tudta kezelni a sejttesttől.

A csapsejtek és a Deiters sejtek nyúlványai morfológiailag hasonlóak olyan szempontból, hogy rendelkeznek egy, a sejttestnél kisebb fejjel. Ezekkel a sejtekkel lefuttatva a programot, majd a ROI-k által mért intenzitásokat görbéken ábrázolva jól látszik, hogy a program sikeresen érzékelt a nyúlvány és a sejttest válaszai közötti különbséget is (13. ábra).

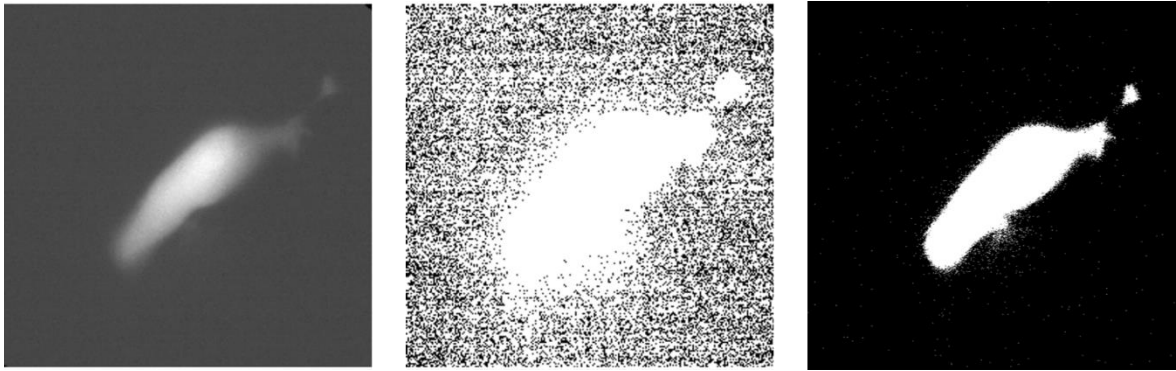


13. ábra: A csapsejt [21] és egy Deiters sejt intenzitásértékei az algoritmussal kiszámolva. A vastagabb piros vonal jelöli a nyúlvány választát, a fekete vékonyabb görbék pedig a sejtesten lévő ROI-k által mért intenzitásokat szemléltetik. Jól látszik a különbség a nyúlvány és a sejtest választai között.

Az idegsejttel kapcsolatban érdemes azonban kiemelni, hogy a sejten lévő ROI-k mérté túl nagy. Ez abból adódott, hogy a kép túl zajos volt, és a kontrasztosítás során ezt nem sikerült eléggé kiküszöbölni. Nagy maradt a szórása annak, hogy az adott oszlopokban melyik sorok tartalmaznak 1-es értéket, így a sejtet nagyobbobnak érzékelt a program, mint amilyen valójában volt. Ennek ellenére a ROI-k így is jó helyre kerültek.

Létezik más kontrasztoszási módszer is, az Otsu például egy általánosan elfogadott eljárás. Ez a módszer is fekete-fehér képekkel dolgozik, a kontrasztosítás pedig a háttér és az előtér elkülönítésére szolgál [24, 25]. Az eljáráshoz az kell, hogy a képből egy bimodiális hisztogramot lehessen alkotni, a küszöbérték így a két csúcs között fog elhelyezkedni [25]. Az Otsu a küszöbértéket úgy határozza meg, hogy a két csúcsot alkotó pixelek csoporton belüli varianciája a lehető legkisebb legyen, ehhez minden lehetséges küszöbértéket kiszámol 0-tól a maximum intenzitásig [24]. Zajos, illetve (közel) unimodiális hisztogramot adó képekre nem működik megfelelően [24].

Kipróbáltuk a program írása során az Otsut is, ám ez lassabbnak bizonyult (mivel nagyon sok küszöbértéket kell ellenőriznie), illetve a mi adatink unimodiális hisztogramot adnak, így nem is teljesül az alkalmazhatósági feltétele. A mi Deiters sejtjeinkről készült képek körülvágásának (a felvételt határoló teljesen fekete, intenzitásértéket nem tartalmazó terület) eltávolítása nagyon torzítja az Otsuhoz az adatokat (14. ábra). A megtalált küszöbértéket a mi módszerünk is ugyanúgy kezeli, mint az Otsu, miszerint a kneedle-vel talált küszöbértéknél kisebb intenzitások 0, a nagyobbak pedig 1-es értéké válnak [24].



14. ábra: Kontrasztolás az Otsu és a mi módszerünket alkalmazva. A bal oldalon az eredeti kép, középen az Otsu kontrasztolás eredménye, a jobb oldalon pedig a Contrast maker lefuttatása utáni állapot látszik.

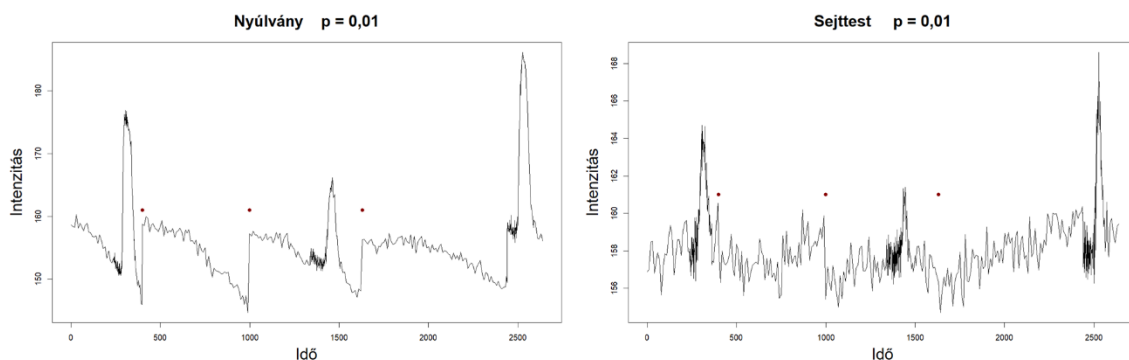
Purkinje sejten [4] is elvégeztük az összehasonlítást, és a mi módszerünk nagyon hasonló eredményt adott, mint az Otsu (15. ábra).



15. ábra: Kontrasztolás Otsuval és a Contrast makerrel. A bal oldalon az eredeti, középen az Otsuval kontrasztolt, a jobb oldalon pedig a Contrast makerrel kontrasztolt purkinje sejt [4] látszik. A két kontrasztolási módszer közel azonos eredményt hozott, az Otsu alkalmazásával kicsit tisztább lett a kép.

Az 3.3.9. fejezetben tárgyalt elcsúszásvizsgálat is működik. A ROI-k újraszámolása nem volt semmilyen negatív hatással az intenzitásokra. Nem generált az áthelyezés új választ, és nem is maradt le egyikről sem (a példa kísérelt során három ingerlés történt: 200, 1300 és 2400 másodpercnél). A 16. ábra szemlélteti a nyúlvány és a sejttest első ROI-ja által kapott intenzitások görbéjén az áthelyezések időpontját. Az elcsúszás kifejezetten a nyúlvány esetében látszik jól, hiszen a nyúlvány méretéből kiindulva sokkal kisebb eltérés is jelentősen beleszól a ROI-n belüli átlagintenzitásba. Jól kivehető, hogy a nyúlvány

elcsúszása emiatt nagyobb súllyal szól bele az áthelyezésbe. A sejttest első ROI-án belül mért intenzitások nem változtak olyan drasztikusan a kísérlet során, mert a sejttest sokkal szélesebb, így többet mozdulhat arrébb mielőtt a program elcsúszást érzékelne. Jobban megnézve az is látszik, hogy a sejttest első ROI-ja alapján nem is kellett volna áthelyezést végezni, így ténylegesen működik a nyúlvány elcsúszásának súlyozása.



16. ábra: A ROI áthelyezések időpontjai (vörös pontok) a nyúlványon és a sejttesten lévő első ROI-val mért átlagintenzitások görbéin ábrázolva. A ROI-k akkor számítottak elcsúszottnak, ha a háttér és az adott ROI intenzitásai nem különböztek szignifikánsan 0,01-es p -értékkel számolva. Azaz a Wilcoxon statisztikai próbából kapott p -érték nagyobb volt, mint 0,01.

5. Megbeszélés és következtetések

Az intracelluláris Ca^{2+} szint vizsgálatával fontos információkat nyerhetünk a sejtek működésével kapcsolatban. Az elmúlt évek során számos olyan módszer született, amely segíti a kutatókat a kalcium ion monitorozásában. Ide tartozik a különböző indikátor molekulák kifejlesztése és a sejtek géntechnológiai módosítása is, mely által a fluoreszcencia segítségével láthatóvá teszik a sejten belüli koncentráció különbségeket. A kísérletek során felvett intenzitás adatok elemzése azonban lassabban fejlődött az elmúlt időben. Sok esetben a felvett adatokat az emberi szubjektivitás is befolyásolja, hiszen a vizsgálni kívánt területeket, a ROI-kat, a kutatók manuálisan határozzák meg, így több kutató eltérő eredményre juthat ugyanannak a kísérletnek az elemzése során [11].

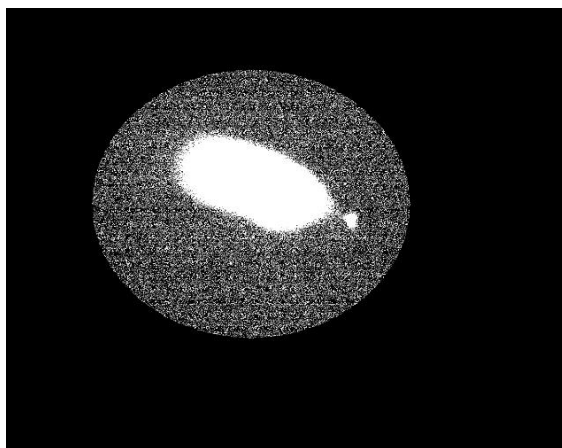
Napjainkban a nagyobb agyterületek idegsejtjeinek automatikus megkeresésére egyre több szoftvert fejlesztenek [8, 11], de az egyedi sejtek vizsgálatának automatizálása elmaradott. Ez ösztönözte egy egyedi, nyúlványos sejtek vizsgálatára alkalmas program

fejlesztését, amelyet a későbbiekben a belső fül Deiters sejtjeinek vizsgálatára szeretnénk felhasználni. Éppen ezért ezt a sejtet tekintettük példa sejtnek a fejlesztés során.

A program a sejtekből készült időbeli felvételsorozatokra képenként a nyúlvánnyal megegyező méretű ROI-kat illeszt, így kiküszöböli a méretekből adódó esetleges műterméket. Ezek mellett minden ötödik rétegnél ellenőrzi a sejt helyzetét, ugyanis egy-egy kísérlet során az változhat.

Összefoglalva, a program megfelelően működik, és képes az elvárásoknak eleget téve objektív elemzést végezni. A szubcelluláris kompartmentek eltérő válaszát is sikeresen lehet vele vizsgálni, ami jelentős szempont. Ezt a példa Deiters sejtünk mellett retinális csapsejtek esetében is sikeresen bizonyítottuk. A 13. ábra jól bemutatja, hogy ezeknél a kompartmenteknél igen jelentős eltérés lehet, így a kutatások során mindenképpen fontos ezeket is vizsgálni, és külön ROI-t helyezni rájuk.

Van azonban egy két tényező, ami a működését hátráltathatja. A kontrasztozáshoz fontos, hogy a kép ne legyen túl zajos, különben a kérdő nem fogja megtalálni azt a küszöbértéket, amivel a sejttest szépen elválasztható lenne a háttértől (17. ábra).



17. ábra: Egy zajos felvétel a kontrasztozás után. A zaj miatt nem sikerült a megfelelő küszöbértéket megtalálni, ezért a kép a küszöb meghatározás és kontrasztozás után is zajos maradt.

A futási idő optimalizálásával sokat foglalkoztunk, és végül sikerült egészen lecsökkenteni. A forgatás telik jelenleg a legtöbb időbe, a jövőben érdemes lehet ennek további optimalizálásával foglalkozni.

A programnak nagy előnye, hogy az R programozási nyelven íródott, mert sok kutatáshoz használnak R-t. Mindenképp könnyebbség, hogy a teljes elemzési folyamat (intenzitás görbék kirajzolása, statisztikai tesztek elvégzése stb.) egy felületen elvégezhető. Jövőbeli cél

azonban, hogy az algoritmus Python nyelvre is át legyen fordítva, így még szélesebb körű elérhetőséget biztosítva.

A program első sorban Deiters sejtek vizsgálatához íródott, ezért más sejtekre felmerülhetnek olyan problémák, amik a sejtek morfológiai különbségeiből adódnak. Éppen emiatt fontos, hogy az algoritmus nyílt forráskódú, mert így a felhasználó testreszabhatja saját igényei szerint.

Összefoglaló

A kalcium a szervezet fontos biogén eleme, amely rendkívül tág szerepkörrel rendelkezik, ezért számos kutatás foglalkozik mechanizmusaival. Gyakran ezen kutatások során kalcium szint monitorozás zajlik, melynek eredményeit számítógépes programok segítségével elemzik. Az elemzéshez hozzátartozik az úgynevezett „region of interest” – rövidítve ROI – felhelyezése is, ami a felvételen vizsgálni kívánt területek kijelölését jelenti. A ROI-k meghatározása azonban manuálisan történik, amire nincs egységes eljárás, megrajzolásuk ezért szubjektív. A szóban forgó programok emellett sokszor nem érhetők el ingyenesen. Fejlesztésük megszűnt, ezáltal elavultak, és támogatásuk fokozatosan lejár az operációs rendszerek újabb verzióival. Az elemzési folyamat nincs bennük automatizálva, elvégzésük ember- és időigényes folyamat. Az eredmények értékelése szempontjából fontos kiemelni, hogy nem nyílt forráskódot használnak, így nem tudni, milyen algoritmusok alapján dolgoznak az adatokkal. Projektünk célja egy nyílt forráskódú, mindenki számára elérhető és tesztre szabható, automatizált elemző program megírása volt, amely objektív módszereket alkalmaz a ROI-k felhelyezéséhez. A program az R programozási nyelven íródott, amely az egyik legelterjedtebb nyelv a biológia tudományában. Fontos szempont volt a lehető legkevesebb R csomag használata. A legtöbb csomag nem kerül az R programmal együtt letöltésre, ezeket a felhasználó maga tölti le egyesével. Mivel az R nyílt forráskódú nyelv, így a csomagok bárki által fejlesztésre kerülhetnek. Előfordulhat, hogy egy frissítés során a csomagba tartozó parancsok valamelyikét átdolgozzák, ami eredményezheti a már megírt program működésbeli változását. A csomagok egyenként történő letöltése ezenfelül lehet lassú, így nem felhasználóbarát. Oka, hogy az R a lassabb programozási nyelvek közé tartozik, ami a kód megírása során kezdetben nehézséget is jelentett. A kísérleti adatok nagysága miatt a futási idő jelentősen megnőtt, ezt idővel azonban sikerült gyorsítani. Elkészült programunkkal sikeresen mértünk különböző típusú, polarizált sejteket. Ez új lehetőségeket kínál majd a kutatóknak, hiszen jelenleg nincs hasonló ingyenes, nyílt forráskódú program, amit ilyen elemzésekre lehetne használni. A továbbiakban cél a program átfordítása Python programozási nyelvre is, a még szélesebb körű alkalmazhatóság érdekében.

Summary

Calcium is an important element of the animal body, therefore it is a highly researched topic. Often these studies involve monitoring calcium levels and analysing the results using computer programs. Selecting the regions of interests – also called ROIs – is a part of the process, which means selecting the areas to be investigated in the recording. However, the ROIs are determined manually, for which there is no standardised procedure, and their definition is hence subjective. Most of these programs are not available for free and their development has stopped, leading to obsolescence. The analysis process is not automated, their implementation is time consuming and human source demanding. It should be highlighted that they are not open-source programs, so it is not known what algorithms are used to deal with the data. Our goal was to develop an open-source, automated analysis program that is freely available to anyone and that uses objective methods to select the ROIs. The coding was done in the R programming language, one of the most used languages among biologists. Minimising the number of R packages used was an important aspect, partly because only a few packages are downloaded with R itself, the others are installed by the customer. Secondly, R is an open-source language so anyone can be a part of its development. The use of many packages increases the possibility that one of them will be updated, therefore changing the outcome of an already coded program. In addition, downloading the packages can be slow, which is less user friendly, and the reason for this is that R is a slower language. This caused difficulties at the beginning, as the size of the experimental data considerably increased the runtime of the code, but this has been improved over time. We were able to measure different types of polarised cells with the finished program. As there is no freely available open-source program like this currently, it will open up new opportunities for researchers. Our aim for the future is to translate this program into Python, another commonly used language in biology.

Köszönetnyilvánítás

Köszönöm témavezetőmnek, Dr. Berekméri Eszternek az odaadását és elhivatottságát aziránt, hogy megfelelően fejlődhessek, és hogy mindig szakít rám időt. Köszönöm, hogy bármilyen kérdésemre kész kielégítő választ adni, és bármilyen problémával kapcsolatban türelemmel segít, ahol kell.

Irodalomjegyzék

1. Berridge MJ, Bootman MD, Lipp P (1998) Calcium - a life and death signal. *Nature* 395:645–648. <https://doi.org/10.1038/27094>
2. Berridge M, Lipp P, Bootman M (1999) Calcium signalling. *Current Biology* 9:R157–R159. [https://doi.org/10.1016/S0960-9822\(99\)80101-8](https://doi.org/10.1016/S0960-9822(99)80101-8)
3. Solovyova N, Verkhratsky A (2002) Monitoring of free calcium in the neuronal endoplasmic reticulum: an overview of modern approaches. *Journal of Neuroscience Methods* 122:1–12. [https://doi.org/10.1016/S0165-0270\(02\)00300-X](https://doi.org/10.1016/S0165-0270(02)00300-X)
4. Grienberger C, Konnerth A (2012) Imaging Calcium in Neurons. *Neuron* 73:862–885. <https://doi.org/10.1016/j.neuron.2012.02.011>
5. Pérez Koldenkova V, Nagai T (2013) Genetically encoded Ca²⁺ indicators: Properties and evaluation. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1833:1787–1797. <https://doi.org/10.1016/j.bbamcr.2013.01.011>
6. IW_Brochure_2007.pdf http://www.imagingworkbench.com/product/IW_Brochure_2007.pdf
7. INDEC BioSystems---Home. <http://www.indecbiosystems.com/>. Accessed 8 Oct 2023
8. Cantu DA, Wang B, Gongwer MW, He CX, Goel A, Suresh A, Kourdougli N, Arroyo ED, Zeiger W, Portera-Cailliau C (2020) EZcalcium: Open-Source Toolbox for Analysis of Calcium Imaging Data. *Frontiers in Neural Circuits* 14:
9. Kaifosh P, Zaremba JD, Danielson NB, Losonczy A (2014) SIMA: Python software for analysis of dynamic fluorescence imaging data. *Frontiers in Neuroinformatics* 8:
10. Kolar K, Dondorp D, Zwiggelaar JC, Høyer J, Chatzigeorgiou M (2021) Mesmerize is a dynamically adaptable user-friendly analysis platform for 2D and 3D calcium imaging data. *Nat Commun* 12:6569. <https://doi.org/10.1038/s41467-021-26550-y>
11. Giovannucci A, Friedrich J, Gunn P, Kalfon J, Brown BL, Koay SA, Taxidis J, Najafi F, Gauthier JL, Zhou P, Khakh BS, Tank DW, Chklovskii DB, Pnevmatikakis EA (2019) CaImAn an open source tool for scalable calcium imaging data analysis. *eLife* 8:e38173. <https://doi.org/10.7554/eLife.38173>
12. Pachitariu M, Stringer C, Dipoppa M, Schröder S, Rossi LF, Dalglish H, Carandini M, Harris KD (2017) Suite2p: beyond 10,000 neurons with standard two-photon microscopy. 061507
13. Parsa A, Webster P, Kalinec F (2012) Deiters cells tread a narrow path—The Deiters cells-basilar membrane junction—. *Hearing Research* 290:13–20. <https://doi.org/10.1016/j.heares.2012.05.006>
14. Zhou W, Jabeen T, Sabha S, Becker J, Nam J-H (2022) Deiters Cells Act as Mechanical Equalizers for Outer Hair Cells. *J Neurosci* 42:8361–8372. <https://doi.org/10.1523/JNEUROSCI.2417-21.2022>
15. R-intro.pdf - <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>
16. Posit. In: Posit. <https://www.posit.co/>. Accessed 10 Oct 2023
17. Tam E (2023) kneedle - <https://github.com/etam4260/kneedle>
18. Satopaa V, Albrecht J, Irwin D, Raghavan B (2011) Finding a “Kneedle” in a Haystack: Detecting Knee Points in System Behavior. In: 2011 31st International Conference on Distributed Computing Systems Workshops. IEEE, Minneapolis, MN, USA, pp 166–171

19. Paeth AW (1990) A FAST ALGORITHM FOR GENERAL RASTER ROTATION. In: Graphics Gems. Elsevier, pp 179–195
20. Gribbon KT, Johnston CT, Bailey DG (2003) A Real-time FPGA Implementation of a Barrel Distortion Correction Algorithm with Bilinear Interpolation
21. Szikra T, Barabas P, Bartoletti TM, Huang W, Akopian A, Thoreson WB, Krizaj D (2009) Calcium Homeostasis and Cone Signaling Are Regulated by Interactions between Calcium Stores and Plasma Membrane Ion Channels. PLoS ONE 4:e6723. <https://doi.org/10.1371/journal.pone.0006723>
22. Klenowski PM, Fogarty MJ, Belmer A, Noakes PG, Bellingham MC, Bartlett SE (2015) Structural and functional characterization of dendritic arbors and GABAergic synaptic inputs on interneurons and principal cells in the rat basolateral amygdala. Journal of Neurophysiology 114:942–957. <https://doi.org/10.1152/jn.00824.2014>
23. Subach OM, Barykina NV, Chefanova ES, Vlaskina AV, Sotskov VP, Ivashkina OI, Anokhin KV, Subach FV (2021) FRCaMP, a Red Fluorescent Genetically Encoded Calcium Indicator Based on Calmodulin from Schizosaccharomyces Pombe Fungus. International Journal of Molecular Sciences 22:111. <https://doi.org/10.3390/ijms22010111>
24. Balarini JP, Nesmachnow S (2016) A C++ Implementation of Otsu’s Image Segmentation Method. Image Processing On Line 6:155–164. <https://doi.org/10.5201/ipol.2016.158>
25. Otsu N (1979) A Threshold Selection Method from Gray-Level Histograms. IEEE Trans Syst, Man, Cybern 9:62–66. <https://doi.org/10.1109/TSMC.1979.4310076>