

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jernej Janežič

Sistem za ocenjevanje poškodb DNK

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: prof. dr. Igor Kononenko

Ljubljana 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 01936/2013

Datum: 03.09.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JERNEJ JANEŽIČ**


Naslov: **SISTEM ZA OCENJEVANJE POŠKODB DNK**
A SYSTEM FOR EVALUATING THE DNA DAMAGE

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Naloga je razviti, implementirati in pretestirati program za ocenjevanje poškodb deoksiribonukleinske kisline (DNK) na podlagi kometnega testa. Kometni test na hiter in enostaven način omogoča merjenje poškodb DNK, pri čemer se iz mikroskopskih posnetkov izračunajo določeni parametri. Kandidat naj v diplomskem delu opiše obstoječe programe za analizo poškodb DNK s kometnim testom ter naj razvije, implementira in pretestira lastni program, ki bo omogočal analizo poškodb DNK z avtomatskim in ročnim določanjem mej kometa. Program naj podpira vse najbolj razširjene formate slik, naj omogoča izbiro parametrov za analizo ter ustrezen izvoz izračunanih podatkov.

Mentor:


prof. dr. Igor Kononenko

Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Jernej Janežič, z vpisno številko **63060087**, sem avtor diplomskega dela z naslovom:

Sistem za ocenjevanje poškodb DNK

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Igorja Kononenka,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 30. avgusta 2013

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju prof. dr. Igorju Kononenku in Venu Kononenku za vso pomoč, nasvete in potrpežljivost pri izdelavi diplomske naloge.

Mojim.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Predvideni cilji diplomskega dela	1
1.3	Pregled diplomskega dela	1
2	Opredelitev problema	3
2.1	Kometni test	3
2.2	Zajem slike kometa	4
2.3	Analiza slike kometa in izračun parametrov	5
2.3.1	Priprava slike in osnovnih parametrov	5
2.3.2	Parametri kometa	7
2.3.3	Neželeni pojavi	9
2.4	Zahteve	10
2.5	Pregled obstoječih orodij za ocenjevanje poškodb DNK	10
2.5.1	CometScore in AutoComet	10
2.5.2	CASP	11
2.5.3	Comet Assay IV	12
3	Uporabljena orodja in metode pri razvoju	13
3.1	Microsoft Visual Studio	13

KAZALO

3.2	.NET Framework	14
3.3	C#	15
3.4	Microsoft WPF	16
4	Razvoj programa za ocenjevanje poškodb DNK	19
4.1	Uporabniški vmesnik	19
4.2	Nalaganje datotek	21
4.3	Ročno označevanje kometov	21
4.4	Avtomatizirano označevanje kometov	22
4.5	Urejanje označenih kometov	24
4.6	Izračun in prikaz rezultatov	25
4.7	Opozorila	25
4.8	Izvoz in shranjevanje rezultatov	26
4.9	Nastavitve	27
5	Zaključek in nadaljnje delo	29
	Slike	31
	Literatura	33

Povzetek

Diplomsko delo obsega razvoj in implementacijo programa za ocenjevanje poškodb DNK na podlagi kometnega testa. V prvem delu je opisan kometni test in njegov potek. Gre za postopek, pri katerem je na hiter in enostaven način mogoča kvantifikacija poškodb molekule DNK. Predstavljeni so parametri, potrebni za natančno analizo kometnega testa. Določene so zahteve za program in predstavljena trenutno dostopna orodja za analizo. Sledi opis tehnologij uporabljenih pri razvoju programa. Uporabili smo programski jezik C#, uporabniški vmesnik pa smo implementirali s pomočjo grafičnega podsistema WPF. V četrtem poglavju je predstavljen razvit program, njegov uporabniški vmesnik in glavne funkcije. Pri razvoju smo se osredotočili predvsem na preprost in pregleden uporabniški vmesnik, ki omogoča hitro in natančno analizo kometov. Program podpira uporabo vseh najbolj razširjenih formatov slikovnih datotek. Omogoča enostavno ročno in avtomatizirano označevanje kometov, izbiro parametrov za analizo in izvoz pridobljenih podatkov v slikovnem formatu ali tabeli. V zaključku so opisani predlogi za nadaljnje izboljšave programa.

Abstract

The thesis describes the development and implementation of the system for the analysis of the comet assay. The first part describes the comet assay. It is a fast and simple process that enables the quantification of the damage of DNA molecules. We present the parameters needed for a precise analysis of the comet assay. The system requirements are presented along with the currently available tools for comet assay analysis. Then we describe the technologies used during the development of the system. The programming language C# is used and the user interface was implemented with the WPF graphical subsystem. Chapter four presents the user interface and the main features of the developed system. The main focus of development was a simple user interface that enables a quick and precise analysis of the comet assay. The system supports the use of all major image formats. It supports simple manual marking as well as automated detection of the comets. The parameters used in the analysis are chosen by the user and the produced results can be exported into an image or a spreadsheet. The possible improvements of the developed system are presented in the conclusion.

Poglavje 1

Uvod

1.1 Motivacija

Pri razvoju novih materialov so potrebna preverjanja, ki nam podajo informacije o potencialnih negativnih vplivih njihove uporabe na živih organizmih. Testiranja sposobnosti povzročitve poškodb DNK (*deoksiribonukleinska kislina*) so pomemben del teh preverjanj. Kometni test je hitra in učinkovita metoda za kvantifikacijo poškodb DNK, pri kateri pa je potrebna zelo specifična programska oprema.

1.2 Predvideni cilji diplomskega dela

Cilj diplomskega dela je razvoj programa, ki z računalniško analizo slike omogoča hitro in zanesljivo kvantifikacijo poškodb DNK na podlagi kometnega testa. Program mora biti enostaven za uporabo, rezultati analize pa morajo biti podani na jasn in pregleden način.

1.3 Pregled diplomskega dela

Prvi del diplomske naloge opisuje potek kometnega testa in postopke, ki so pri njem uporabljeni. V tretjem poglavju so opisana orodja in tehnologije,

ki smo jih uporabili pri razvoju programa. Četrto poglavje opisuje lastnosti razvitega programa in načine, s katerimi smo rešili probleme. V zadnjem delu diplomske naloge so povzete izdelane funkcije in predlogi za nadaljnji razvoj.

Poglavje 2

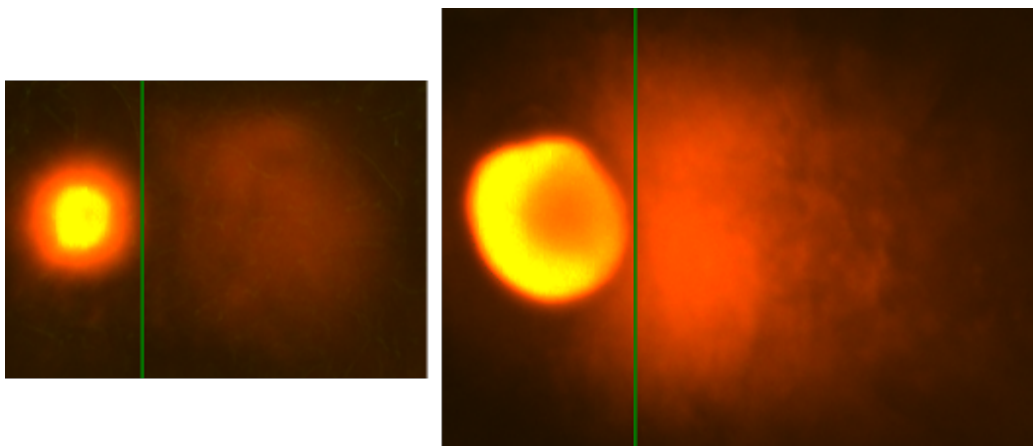
Opredelitev problema

To poglavje je povzeto po delu dr. Barbare Vilhar, "Help! There is a comet in my computer" [4].

2.1 Kometni test

Kometni test (*angl. comet assay*) nam omogoča zaznavanje prelomov molekule DNK, zato predstavlja hitro in učinkovito metodo za kvantifikacijo poškodb DNK.

Celice vklopimo v agarozni gel, ki ga med postopkom elektroforeze izpostavimo električnemu toku. Molekule DNK so negativno nabite, zato se v električnem polju pomaknejo proti pozitivno nabiti elektrodi. Večje, nepoškodovane molekule se skozi gel pomikajo počasneje kot manjši, poškodovani delci. Po elektroforezi gel s celicami obarvamo s fluorescentnim barvilom, ki se veže na DNK celic, pri čemer je pomembno dejstvo, da je razmerje med količino DNK in količino vezanega barvila *linearno*. Ko so celice obarvane, gel osvetlimo s svetlobo specifične valovne dolžine (odvisno od uporabljenega fluorescentnega DNK-veznega barvila), pri čemer barvilo vezano na DNK odda svetlobo višje valovne dolžine. Da pridobimo sliko le zelene valovne dolžine svetlobe, uporabimo set ustreznih *svetlobnih filtrov*, ki do vzorca prepuščajo le valovno dolžino eksitacije barvila, do uporabnika (de-



Slika 2.1: Primer kometa z majhnimi (levo) in velikimi poškodbami (desno).

tektorja) pa le oddano fluorescentno svetlobo. Količina vezanega barvila je linearna s količino oddane svetlobe, torej je na močnejše osvetljenih predelih slike večja količina barvila. Količina barvila pa je linearna s količino DNK, zato lahko sklepamo, da je količina DNK linearna s količino zaznane svetlobe pri zajemu slike. Bolj osvetljeni predeli slike torej vsebujejo več DNK.

Problem ocenjevanja poškodb DNK lahko zato preslikamo na problem ocenjevanja intenzitete osvetljenosti območij slike. Nepoškodovane molekule se med testom ne premikajo daleč in ostanejo strnjene, zato so na sliki območja z njimi običajno najsvetlejša. Predstavljajo *glavo* kometa, poškodovani delci pa so razpršeni, zato so deli slike, na katerih se nahajajo, manj svetli - predstavljajo *rep* kometa.

Primer obarvanih molekul DNK je viden na sliki 2.1.

2.2 Zajem slike kometa

Zaznana svetloba je zelo šibka. Barvilo namreč svetlobo oddaja v vse smeri, vidna pa je le svetloba, ki pade v objektiv mikroskopa. Poleg tega je večina fluorescentnih barvil fotosenzitivnih (če so izpostavljena svetlobi, pričnejo kemijsko razpadati). Za lažje in bolj natančno opazovanje je zato potrebno

zajem slike izvajati v zatemnjenem prostoru.

Zajem slike izvajamo z digitalno kamero (nameščeno na mikroskop), ki za zaznavanje svetlobe uporablja množico majhnih, pravokotnih, na svetlobo občutljivih elementov. Vsak izmed teh elementov pretvarja intenziteto zaznane svetlobe v digitalen zapis, običajno v števila. Nižje število predstavlja manjšo, višje število pa večjo intenziteto svetlobe. Ob zapisu zaznanih vrednosti se zabeležijo tudi koordinate elementa, zato je prebrana števila mogoče obravnavati kot točke na digitalni sliki (*angl. pixels*).

Običajno je zajem slike ločen od analize: pripravimo kometni test, ga pod mikroskopom slikamo in sliko prenesemo na računalnik, kjer z orodji za ocenjevanje kometov izvajamo analizo. Obstajajo pa tudi orodja, ki analizo opravljajo *v živo*, v tem primeru so kometi analizirani sproti med mikroskopiranjem.

Kvaliteta zajete slike je za kometni test bistvenega pomena, saj z morebitno izgubo informacij zmanjšujemo natančnost analize in izračuna parametrov.

2.3 Analiza slike kometa in izračun parametrov

2.3.1 Priprava slike in osnovnih parametrov

Prvi korak pri analizi je pretvarjanje barvne slike v sivinsko. Osredotočamo se na intenziteto svetlobe in ne na njeno valovno dolžino, zato so slike v sivinskem načinu za analizo bolj primerne.

Za izračun parametrov kometa je potrebno iz slike razbrati nekaj osnovnih podatkov. Skupna intenziteta svetlobe je vsota vseh vrednosti slikovnih točk kometa:

$$total_{comet}[x] = \sum_{y=0}^{height} Img[x, y] \quad (2.1)$$

$$Total_{comet} = \sum_{x=0}^{width} total_{comet}[x] \quad (2.2)$$

$Img[x, y]$ predstavlja vrednost slikovne točke v stolpcu x in vrstici y sivinske slike Img . Če z vrednostjo $head_{width}$ določimo lokacijo meje med glavo in repom kometa na osi x , pa lahko skupni intenziteti glave in repa, $total_{head}$ in $total_{tail}$ definiramo z:

$$Total_{head} = \sum_{x=0}^{head_{width}} total_{comet}[x] \quad (2.3)$$

$$Total_{tail} = \sum_{x=head_{width}+1}^{width} total_{comet}[x] \quad (2.4)$$

Ozadje pa nikoli ni popolnoma črno (torej na sliki nima vrednosti 0), zato ga je potrebno upoštevati pri izračunu (govorimo o t.i. *integrirani svetlobi*). Da bi lahko prišli do pravih vrednosti parametrov, moramo vpliv svetlobe ozadja (njegovo intenziteto) od skupne intenzitete *odšteti*. Zato pri zajemu slike ne zajamemo le območja, na katerem se nahaja komet, ampak tudi pas točk pravokotno nad (ali pod) kometom (slika 2.2, A). Povprečna intenziteta avg_{bg} pasu ozadja Img_{bg} z višino $height_{bg}$ v stolpcu x je:

$$avg_{bg}[x] = \frac{\sum_{y=0}^{height_{bg}} Img_{bg}(x, y)}{height_{bg}} \quad (2.5)$$

Skalirana intenziteta ozadja $scaled_{bg}$ za višino $height$ v stolpcu x pa:

$$scaled_{bg}[x] = avg_{bg}[x] \times height \quad (2.6)$$

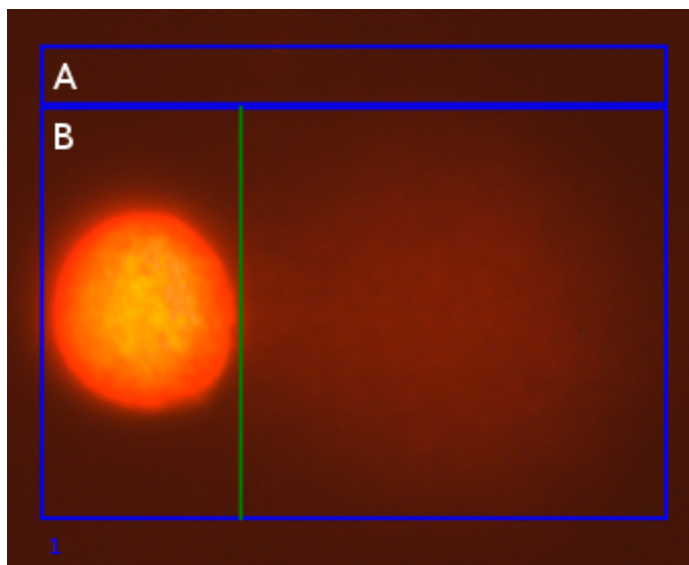
Popravljen skupna intenziteta svetlobe kometa je torej:

$$corrected_{comet}[x] = total_{comet}[x] - scaled_{bg}[x] \quad (2.7)$$

$$Corrected_{comet} = \sum_{x=0}^{width} corrected_{comet}[x] \quad (2.8)$$

Popravljeni intenziteti glave in repa pa:

$$Corrected_{head} = \sum_{x=0}^{head_{width}} corrected_{comet}[x] \quad (2.9)$$



Slika 2.2: **A:** Pas točk, uporabljen za izračun vpliva ozadja. **B:** Območje kometa.

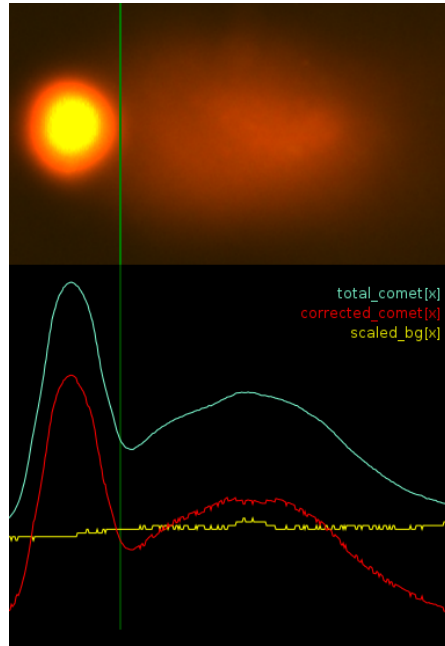
$$Corrected_{tail} = \sum_{x=head_{width}+1}^{width} corrected_{comet}[x] \quad (2.10)$$

2.3.2 Parametri kometa

Za ocenjevanje poškodb uporabljamo nekaj različnih parametrov:

- **Profil kometa** - graf, ki prikazuje vsote intenzitet stolpcev kometa (slika 2.3). Običajno prikazujemo intenziteto kometa $total_{comet}[x]$, popravljen intenziteto kometa $corrected_{comet}[x]$ in skalirano intenziteto ozadja $scaled_{bg}[x]$.
- **Dolžina repa** - $TailLength$ je dolžina repa kometa v mikrometrih.
- **Odstotek DNK v glavi** - razmerje med skupno intenziteto glave in skupno intenziteto celotnega kometa.

$$HeadDNA = \frac{Corrected_{head}}{Corrected_{comet}} \quad (2.11)$$



Slika 2.3: Primer profila kometa

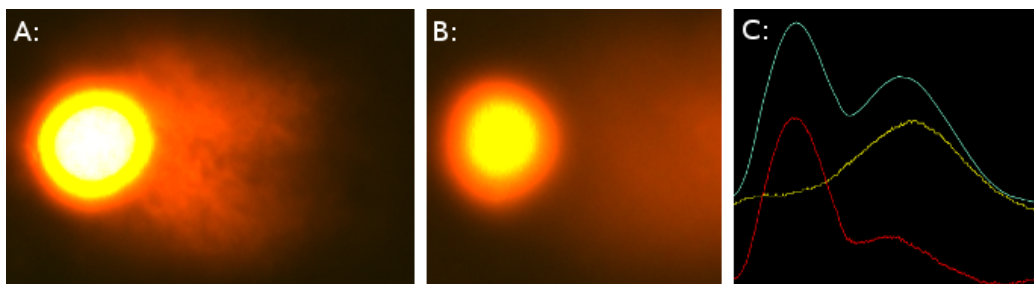
- **Odstotek DNK v repu** - razmerje med skupno intenziteto repa in skupno intenziteto celotnega kometa.

$$TailDNA = \frac{Corrected_{tail}}{Corrected_{comet}} \quad (2.12)$$

- **Tail extent moment** - produkt dolžine repa in odstotka DNK v repu. Enota TEM je mikrometer.

$$TailMoment = TailLength \times TailDNA \quad (2.13)$$

- **Olive tail moment** - produkt odstotka DNK v repu in razdalje med centroma intenzitet glave in repa na x osi. Enota OTM je mikrometer. Center intenzitete glave je določen kot točka, za katero velja, da je polovica popravljene intenzitete $Corrected_{head}$ na njeni levi, polovica pa na desni strani. Je torej najvišja vrednost x , pri kateri še velja pogoj $\sum_{i=0}^{i=x} corrected_{head}[i] \leq \frac{Corrected_{head}}{2}$. Enako velja za center intenzitete



Slika 2.4: **A:** Komet s previsoko osvetljenostjo. **B:** Komet z odrezanim repom. **C:** Profil kometa z neenakomernim ozadjem (graf $scaled_{bg}[x]$ je izrisan z rumeno barvo).

repa.

$$OTM = TailDNA \times (centroid_{tail} - centroid_{head}) \quad (2.14)$$

2.3.3 Neželeni pojavi

Pri analizi slik kometov lahko naletimo na nekatere neželene pojave.

Neenakomerno ozadje je vidno na profilu kometa, če je površina, uporabljena za izračun intenzitete ozadja, neenakomerno osvetljena (slika 2.4, C). Neenakomerno ozadje je najpogosteje posledica nečistoč v slikanem gelu ali prisotnosti drugega kometa v ozadju. Želimo, da imajo vrednosti skupnih intenzitet ozadja čim manjša odstopanja od povprečja, graf $scaled_{bg}[x]$ naj bi bil zato čim bolj podoben vodoravni črti.

Pozorni moramo biti tudi na vrednosti slikovnih točk, saj za njih obstajata dve omejitvi. V primeru, da so na sliki kometa prisotne točke z vrednostjo 0 ali 255, govorimo o *nasičenosti* (angl. *saturation*). Kot že omenjeno, želimo intenziteto odbite svetlobe opisati z maksimalnim številom vrednosti, vendar pa robne vrednosti 0 (pomanjkanje svetlobe, angl. *underexposure*) in 255 (previsoka osvetljenost, angl. *overexposure*, prikazana na sliki 2.4, A) pomenijo napake pri zajemu slike in posledično napačne rezultate pri analizi.

Pri zajemu slike kometa moramo zagotoviti, da je komet na njej prisoten v celoti. V primeru da z robom slike "odrežemo" komet, lahko izgubimo po-

membren del informacij (slika 2.4, B). Takšne slike niso primerne za analizo.

2.4 Zahteve

Okvirne zahteve za program:

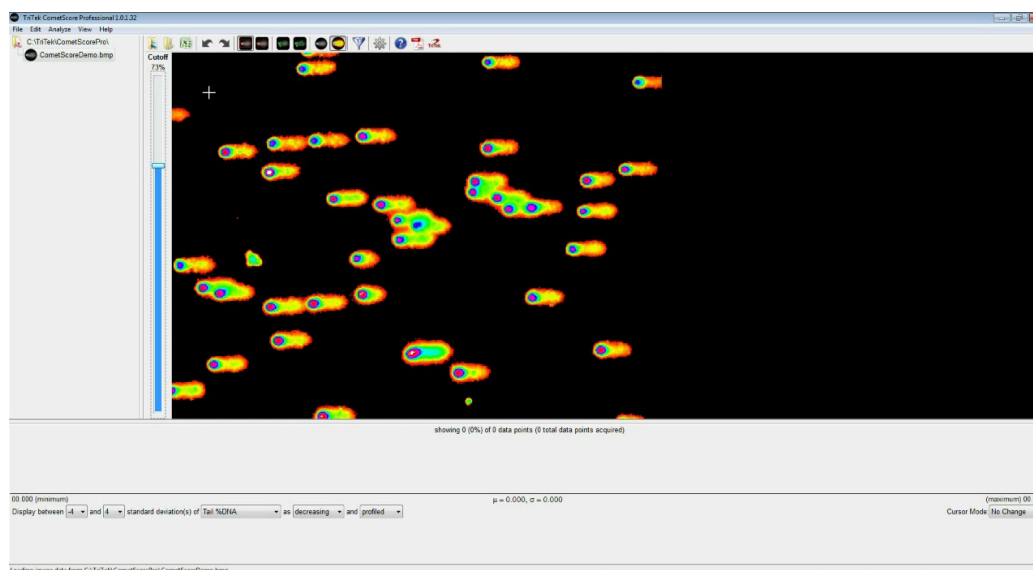
- enostaven in pregleden uporabniški vmesnik,
- ročno označevanje kometov,
- avtomatizirano označevanje kometov,
- premikanje, spreminjanje velikosti, brisanje kometov,
- izvoz podatkov v Excell (CSV),
- izvoz slik in profilov kometov,
- omejevanje velikosti in kontrasta kometov,
- izbira parametrov za izračun,
- opozorila pri kometih z neželenimi pojavi.

2.5 Pregled obstoječih orodij za ocenjevanje poškodb DNK

2.5.1 CometScore in AutoComet

Orodji CometScore in AutoComet, razviti pri podjetju TriTek Corp, sta komercialni orodji za ocenjevanje poškodb DNK na osnovi kometnega testa.

CometScore je na voljo v dveh različicah, CometScore in CometScore Pro. Cena licence za prvo znaša v času pisanja tega dela 95\$, za drugo pa 1295\$. Osnovna različica omogoča ročno označevanje kometov, ročno segmentacijo ozadja in izvoz podatkov v tabelo. Različica Pro je precej bolj zmogljiva:



Slika 2.5: Uporabniški vmesnik CometScore Pro

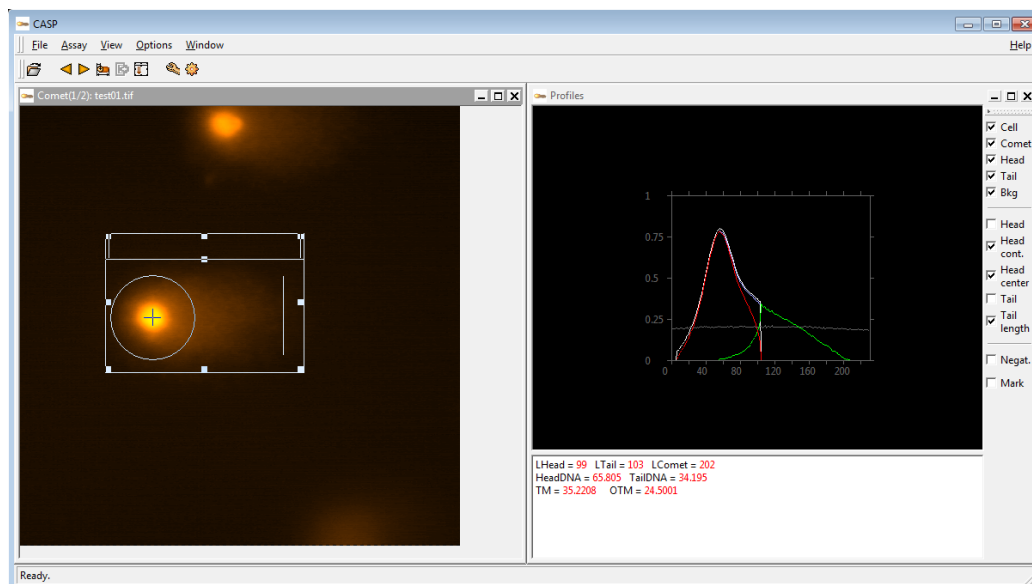
ročno ocenjevanje kometov je izboljšano in enostavnejše (z le enim klikom, brez označevanja mej kometa), omogoča tudi avtomatizirano detekcijo in ocenjevanje kometov, določeno filtriranje šuma ter prikaz in izvoz histogramov slik. Slika 2.5 prikazuje uporabniški vmesnik CometScore Pro.

Aplikacija AutoComet je najzmogljivejša, cena za licenco v času pisanja tega dela na spletni strani ni navedena. Poleg vseh zmogljivosti CometScore Pro omogoča tudi avtomatsko segmentacijo ozadja, podpira več-uporabniški način dela in hranjenje podatkov v podatkovni bazi.

Kot je razvidno iz opisov, gre za zelo močna orodja, njihova glavna pomanjkljivost pa je cena licence za uporabo.

2.5.2 CASP

CASP je odprtokodna programska oprema, na voljo pod licenco GNU. Podpira uporabo različnih formatov slik, nalaganje več slik hkrati in shranjevanje ter izvoz rezultatov. Označevanje kometov je ročno, vendar pa program avtomatsko locira center in radij glave kometa.



Slika 2.6: Uporabniški vmesnik CASP

Glavna pomankljivost programa je v nekoliko nerodnem uporabniškem vmesniku, ki ne omogoča istočasnega označevanja večjega števila kometov na eni sliki. Uporabniški vmesnik CASP je viden na sliki 2.6.

2.5.3 Comet Assay IV

Comet Assay IV je komercialno orodje, razvito pri podjetju Perceptive Instruments. Omogoča ocenjevanje kometov s t.i. *single click automatic scoring* metodo, pri kateri uporabnik z eno potezo označi komet, program pa samodejno izračuna parametre za ocenjevanje. Comet Assay IV deluje *v živo*, torej ocenjevanje kometov poteka že med samim opazovanjem pod mikroskopom, brez vmesnega koraka zajema slike in njenega prenosa na računalnik. Podpira izvoz podatkov in njihovo analizo, shranjevanje slik obravnavanih kometov in povezovanje s podatkovnimi bazami Oracle.

Poglavje 3

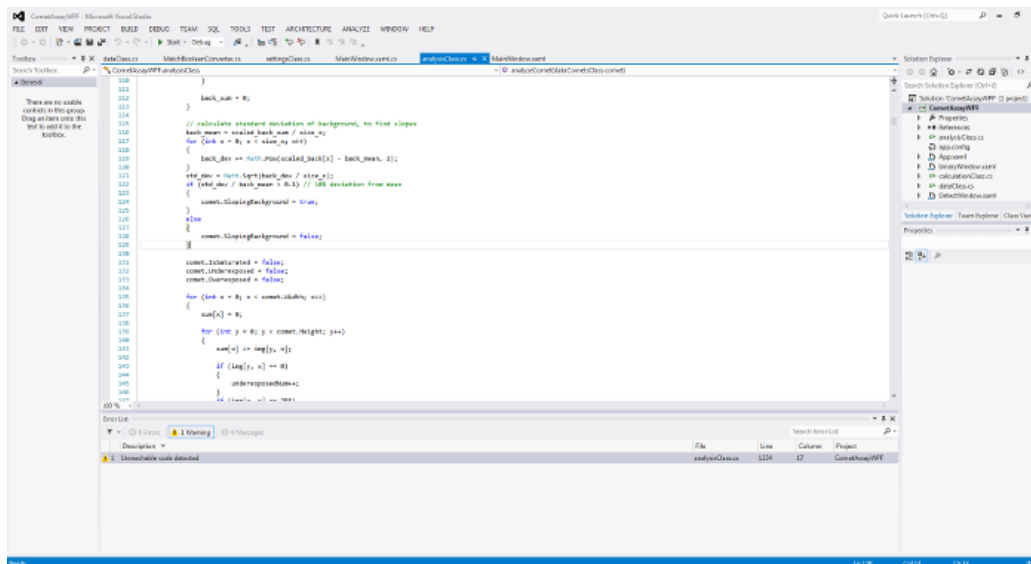
Uporabljena orodja in metode pri razvoju

3.1 Microsoft Visual Studio

Microsoft Visual Studio [6] je integrirano razvojno okolje (*angl. integrated development environment - IDE*), razvito pri podjetju Microsoft. Vključuje podporo za razvoj z različnimi programskimi jeziki, kot so C/C++, VB.NET, C# in nekateri drugi. Omogoča enostavno načrtovanje uporabniških vmesnikov, spletnih aplikacij, razredov in podatkovnih baz. Aplikacija ima tri glavne dele: urejevalnik programske kode (*angl. code editor*), razhroščevalnik (*angl. debugger*), in oblikovalec (*angl. designer*).

Urejevalnik programske kode med drugim vključuje obarvanje sintakse, pomoč pri pisanju izrazov (code completion, autocomplete), zapiranje blokov programske kode, iskanje in prevajanje v ozadju. Pri prevajanju v ozadju se programska koda prevaja kar med samim pisanjem z namenom iskanja sintaktičnih in drugih napak, na katere urejevalnik opozarja z rdečim podčrtovanjem problematičnih delov.

Vgrajeni razhroščevalnik deluje na nivoju izvorne kode in na strojnem nivoju. Omogoča razhroščevanje aplikacij, napisanih v vseh jezikih, podprtih v Visual Studio, možno pa je tudi razhroščevanje že tekočih procesov. Če



Slika 3.1: Uporabniški vmesnik Microsoft Visual Studio

izvirna koda za trenutni proces ni na voljo, z uporabo *disassemblerja* priskrbi kodo v zbirniku (*angl. assembly language*). Podprto je tudi razhroščevanje več-nitnih programov.

Oblikovalec združuje široko paleto orodij za oblikovanje uporabniških vmesnikov in spletnih aplikacij ter za načrtovanje razredov in podatkovnih baz.

Uporabniški vmesnik okolja je prikazan na sliki 3.1.

3.2 .NET Framework

.NET Framework [11] je programsko ogrodje, razvito pri podjetju Microsoft. Vključuje obširno knjižnico in omogoča jezikovno interoperabilnost (vsak jezik lahko uporablja kodo, napisano v drugih jezikih) za nekaj različnih jezikov.

Razvoj aplikacij za platformo Windows poteka s kombiniranjem lastne programske kode z ogrodjem .NET in drugimi knjižnicami. Programi, napisani v ogrodju .NET, se izvajajo v programskem okolju CLR (Common

Language Runtime), ki skrbi za varnost, upravljanje s pomnilnikom in obravnavanje izjem.

Knjižnica BCL (*Base Class Library*) vključuje majhno podmnožico razredov celotne knjižnice .NET in služi kot osnovni API za CLR. Omogoča razvoj uporabniških vmesnikov, povezovanje s podatkovnimi bazami, uporabo kriptografije, numerične algoritme in omrežno komunikacijo. Knjižnica FCL (*Framework Class Library*) je nadmnožica BCL in vključuje vse razrede knjižnice .NET. FCL je v primerjavi s standardnimi knjižnicami drugih jezikov, kot npr. C++, zelo velika, zato je bolj primerljiva s standardno knjižnico jezika Java. Knjižnica .NET in CLR skupaj gradita ogrodje .NET.

Ogrodje vsebuje orodja za pripravo enostavnih namestitvenih programov, ki preprečujejo težave s poseganjem v že nameščeno programsko opremo. V določeni meri ogrodje .NET skrbi tudi za varnost, saj opozarja in preprečuje izkoriščanje nekaterih varnostnih ranljivosti, kot so prekoračitve pomnilnika (*angl. buffer overflows*).

Ogrodje .NET je na voljo tudi v različicah za uporabo na mobilnih in vgrajenih (*angl. embedded*) napravah.

3.3 C#

C# [5, 3] je večnamenski programski jezik, razvit v okviru .NET pri Microsoft. Je strogo tipiziran jezik, ki omogoča deklarativno, kot tudi imperativno in funkcijsko programiranje. Namenjen je objektno-orientiranemu razvoju, možen pa je tudi proceduralni razvoj.

Gre za enostaven jezik s približno 80 ključnimi besedami, čigar razvoj se je pričel leta 2000. Glavni načrtovalec in vodja razvojne ekipe jezika je Anders Hejlsberg. Čeprav naj bi bili programi napisane v C# ekonomični s porabo pomnilnika in procesne moči, razvijalci jezika niso nameravali tekmovali s hitrostjo in porabo jezikov, kot sta C in zbirnik. Jezik in vse njegove implementacije naj bi podpirale principe, kot so strogo preverjanje tipov, preverjanje meja polj, zaznavanje uporabe neiniciliziranih spremenljivk in

garbage collection.

Omogoča sprotno pisanje dokumentacije v obliki XML, ki olajša oblikovanje klasične ali spletne oblike dokumentacije razvitega programa.

V času pisanja tega dela je najnovejša različica jezika C# 5.0.

3.4 Microsoft WPF

WPF (*angl. Windows Presentation Foundation*) [7, 2] je grafični podsistem, uporabljen pri razvoju modernih aplikacij za operacijski sistem Windows. Namesto starejšega vmesnika GDI (Graphics Device Interface) uporablja vmesnik DirectX. WPF ločuje uporabniški vmesnik in poslovno logiko.

Uporaba DirectX namesto starejšega vmesnika omogoča izkoriščanje strojnega grafičnega pospeševanja (*angl. hardware acceleration*). DirectX je prilagojen modernim grafičnim karticam in zato odlično izkorišča njihove zmogljivosti, zato je v okviru WPF mogoče razvijati tudi zelo kompleksne, 3-dimenzionalne programe.

Pomembna lastnost WPF je tudi *resolucijska neodvisnost*. To pomeni, da za enote velikosti posameznih elementov ne uporablja slikovnih točk, temveč enote neodvisne od naprave (*angl. device-independent units*). Takšna enota je enaka 1/96 palca (2,54 cm). Koliko slikovnih točk bo torej uporabljenih za prikaz elementa velikosti ene enote WPF pa je odvisno od sistemske nastavitve gostote točk (*angl. DPI - dots per inch*). V praksi to pomeni, da bo element, čigar velikost je definirana z 1 palcem, prikazan v enaki fizični velikosti na zaslonu z ločljivostjo 1200x900 in zaslonu z ločljivostjo 800x600.

Za definiranje in povezovanje elementov grafičnega vmesnika uporablja XAML [9], jezik baziran na XML [10]. Glavna prednost uporabe XAML za razvoj grafičnega vmesnika je v popolni deklarativnosti jezika. Razvijalec lahko opiše komponente brez uporabe proceduralnega programiranja, kar močno olajša vklop oblikovalcev v razvojni cikel programa. Uporaba XAML omogoča ločitev podatkovnega modela in grafičnega izgleda. Primer kode v jeziku XAML je prikazan na sliki 3.2.

```
<Window x:Class="WindowsApplication1.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="300">

  <Grid>
    <Button>Hello world</Button>
  </Grid>
</Window>
```

Slika 3.2: Primer kode v jeziku XAML

Data binding sicer ni novost WPF, vendar močno poenostavi razvoj grafičnega vmesnika vsakega programa. Omogoča povezovanje podatkovnih struktur z grafičnim vmesnikom na hiter in enostaven način, z minimalno količino programske kode. Grafičnemu elementu (vnosno polje, slika, tabela) določimo vir podatkov, za prikaz, osveževanje in zapisovanje novih vrednosti pa poskrbi kar WPF. Vključuje tudi validacijo vnosnih polj, kar še dodatno poenostavi razvoj na tem področju.

Poglavje 4

Razvoj programa za ocenjevanje poškodb DNK

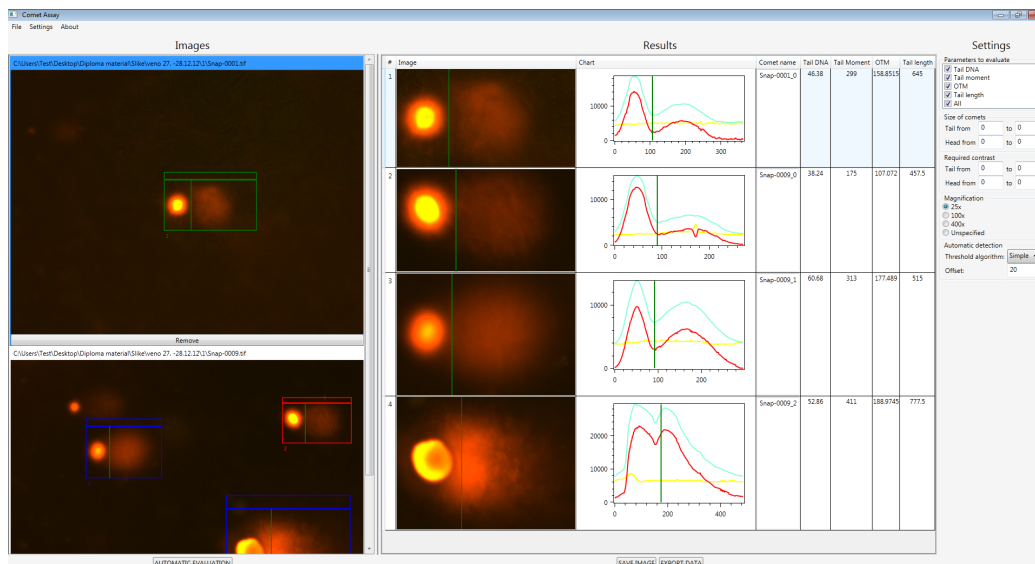
Razvoj je potekal od zgoraj navzdol, kar pomeni, da smo najprej pripravili uporabniški vmesnik, nato pa mu postopoma dodajali funkcionalnosti.

4.1 Uporabniški vmesnik

Uporabniški vmesnik (slika 4.1) smo zasnovali tako, da bo za uporabnika kar se da enostaven in pregleden. Želeli smo ohraniti pozitivne lastnosti obstoječih orodij za ocenjevanje in se izogniti njihovim pomanjkljivostim. Postavitev je zasnovana na postavitvi orodja CASP, vendar smo jo za večjo preglednost precej spremenili. Glavno okno programa smo razdelili na 3 dele:

- seznam slik,
- seznam obravnavanih kometov,
- hitre nastavitve.

Širina dela hitrih nastavitvev je konstantna, medtem ko je preostali prostor enakomerno razdeljen med seznam slik in seznam obravnavanih kometov. Med seznamoma je kontrola, ki omogoča spreminjanje razmerja velikosti med njima, tako da je vsak seznam mogoče po potrebi tudi razširiti.



Slika 4.1: Uporabniški vmesnik programa

Seznam slik na levi strani prikazuje vse naložene slike. Nad vsako sliko je prikazano ime datoteke in njena velikost v slikovnih točkah. Pod sliko je gumb, ki omogoča izbris slike s seznama, z izbrisanjem slike pa se izbrisejo tudi vsi označeni kometi. Na sliki so označeni kometi prikazani s pravokotniki, pod vsakim pravokotnikom pa je izpisana tudi zaporedna številka kometa. Pod seznamom slik je gumb z napisom "Automatic evaluation", ki sproži avtomatizirano označevanje kometov.

Vsak komet (označen ročno ali z avtomatskim označevanjem) je prikazan na *seznamu obravnavanih kometov*. V prvem stolpcu seznama je vidna zaporedna številka kometa, sledijo pa slika kometa, njegov profil, ime in vsi izbrani parametri za ocenjevanje. Pod seznamom kometov sta dva gumba z napisoma "Save image" in "Export data". S prvim shranjujemo slike kometov in njihove profile, z drugim pa izvoz vrednosti parametrov za ocenjevanje v tabelo.

Nekaj nastavitvev programa je prikazanih v *hitrih nastavitvah*, ki se nahajajo na desni strani. Tukaj lahko uporabnik izbere parametre za ocenjevanje, omejitve glede velikosti in kontrasta kometa, povečavo mikroskopa in algo-

ritme filtriranja pri avtomatiziranem označevanju kometov. Za večjo delovno površino je hitre nastavitve mogoče tudi odstraniti.

4.2 Nalaganje datotek

Ob zagonu programa sta seznama slik in obravnavanih kometov prazna, uporabnik pa datoteke slik v program naloži na dva načina. Z izbiro *File* → *Load images* v meniju programa se odpre okno za izbor, možno je nalaganje večih datotek hkrati. Nalaganje pa je mogoče tudi s tehniko "drag and drop", pri kateri zelene datoteke označimo in z miško potegnemo v seznam slik.

Podprti so vsi glavni formati slikovnih datotek: JPEG, PNG, TIF, BMP. Za vsako datoteko se ob nalaganju v ozadju generira in v pomnilniku shrani objekt, ki med drugim vsebuje tudi njeno sivinsko različico. Ta je kasneje uporabljena predvsem pri avtomatiziranem označevanju kometov.

4.3 Ročno označevanje kometov

Na vsaki naloženi sliki je mogoče ročno označevanje kometov. Pri načrtovanju tega dela smo se zgledovali po aplikaciji CometScore.

Komet lahko na sliki označimo tako, da levi miškin gumb pritisnemo in držimo, kazalec miške pa premikamo in tako narišemo pravokotnik, ki označuje komet. Ko gumb spustimo, se nad pravokotnikom prikaže še površina ozadja, pod pravokotnikom pa zaporedna številka kometa. V primeru, da smo komet označili blizu zgornjega roba slike, in zato za površino ozadja nad njem ni prostora, se le ta samodejno ocenjuje pod kometom. Območje, ki smo ga s pravokotnikom označili, je prikazano v seznamu obravnavanih kometov, samodejno je zgenerirano ime kometa, izračunani pa so tudi parametri za ocenjevanje.

Določanje meje med glavo in repom kometa je zelo enostavno. Z miškinim kazalcem se premaknemo na mesto, kjer smo ocenili, da se meja nahaja. Če meja kometa še ni bila določena, pokončna črta (ki določa delitev), sledi

kazalcu miške. S klikom se meja fiksira. Kasneje lahko mejo premaknemo tako, da nanjo kliknemo, gumb držimo in mejo z miško potegnemo na zeleno mesto.

4.4 Avtomatizirano označevanje kometov

Ročno označevanje kometov je lahko časovno zelo potratno, zato smo želeli v aplikacijo vključiti tudi funkcijo avtomatiziranega označevanja. Ko so slike s kometi naložene, lahko označevanje sprožimo s klikom na gumb "Automatic evaluation". Označevanje se izvaja v ozadju, uporabniku pa se prikaže okno, ki prikazuje napredek.

Prvi korak pri avtomatiziranem označevanju kometov je *segmentacija slike*, torej ločevanje ozadja od regij, ki so za našo analizo zanimive. Za segmentacijo smo uporabili metodo upragovanja: sivinsko sliko Img pretvorimo v binarno Img_{bin} na podlagi praga t .

$$Img_{bin}(x, y) = \begin{cases} 0 & Img(x, y) < t \\ 1 & \text{sicer} \end{cases} \quad (4.1)$$

Če je vrednost sivinske slike v točki (x, y) manjša od vrednosti praga, jo v binarno sliko zapišemo kot 0 (črna), v nasprotnem primeru pa 1 (bela). Vrednost praga t pa lahko določimo na več različnih načinov. V programu smo vključili dva algoritma za določanje praga:

- Otsujeva metoda [8], ki na podlagi histograma išče vrednost, ki bo sliko razdelila na dva dela z minimalno varianco.
- Ročna metoda, ki s pomočjo histograma slike določi vrednost ozadja (gre za vrednost z najvišjim stolpcem v histogramu), nato pa tej vrednosti prišteje odmik (*angl. offset*). Odmik določi uporabnik.

Za najbolj učinkovito se je izkazala ročna metoda z nizkim odkom.

Segmentaciji slike sledi *filtriranje* z morfološkim alternirajočim sekvenčnim filtrom (*angl. morphological alternating sequential filter*), ki na pridobljeni binarni sliki večkrat izvede morfološko odpiranje in zapiranje slike, s

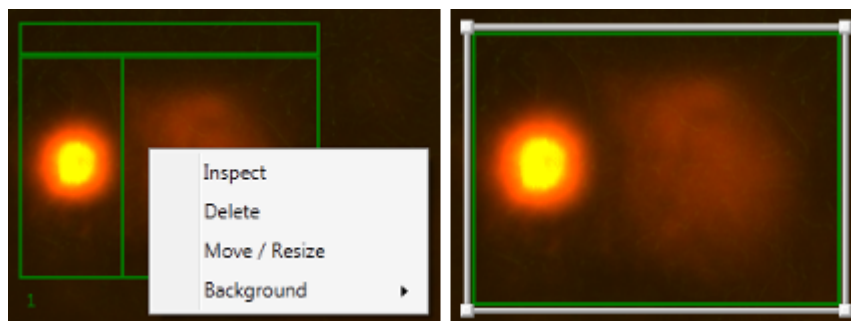
čimer zgladi robove in zapolni luknje v belih regijah. Filtriranje je pomemben del zaznavanja, saj z njim v veliki meri odstranimo nepravilnosti, ki so nastale pri upragovanju slike. Med postopkom filtriranja je potrebna večkratna iteracija skozi vse točke slike, zato gre za časovno potratno operacijo. Fazo filtriranja je mogoče izklopiti v nastavitvah programa, s čimer zaznavanje močno pospešimo, vendar za ceno odpornosti algoritma na šumne podatke.

Po končanem filtriranju iz slike "izrežemo" detektirane objekte z uporabo Moorovega postopka za sledenje mej [1]. Vsak objekt predstavlja komet, ki ga je potrebno analizirati. V primeru, da se objekt nahaja blizu zgornjega roba slike in zato ni mogoče obravnavati dovolj širokega pasu ozadja, se okvir ozadja samodejno prestavi *pod* okvir kometa.

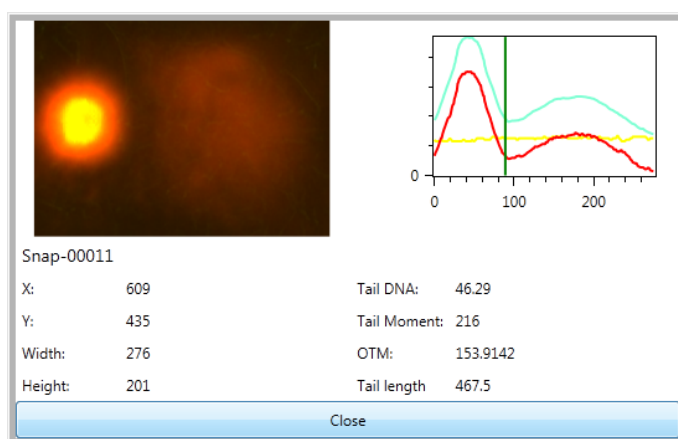
Preden lahko izračunamo parametre za ocenjevanje pa moramo določiti mejo med glavo in repom kometa. Za rešitev tega problema smo uporabili profil kometa. Kometi so v kometnem testu orientirani tako, da je glava kometa vedno na levi strani, zato je center glave najpogosteje na mestu z najvišjo vrednostjo $corrected_{comet}[x]$ (najsvetlejši stolpec slikovnih točk slike). Glava kometa je okrogle ali eliptične oblike, zato predvidevamo, da je razdalja med levim robom slike kometa in točko z najvišjo vrednostjo $corrected_{comet}[x]$ enaka polovici širine glave. Meja med glavo in repom se torej nahaja v točki $2 \times x$.

Izračunu meje med glavo in repom kometa sledi izračun parametrov za ocenjevanje, s tem pa je avtomatizirano označevanje kometov zaključeno.

Rezultate avtomatiziranega označevanja je po končanem postopku vedno potrebno pregledati in po potrebi ročno popraviti. Napakam pri avtomatiziranem zaznavanju se ni mogoče popolnoma izogniti, saj ocenjujemo slike bioloških vzorcev, ki so zelo raznoliki, na slikah pa se lahko pojavljajo artefakti (kristali barvila, ostanki odmrlih celic, prah in podobno).



Slika 4.2: **Levo:** dodatne funkcije kometa. **Desno:** oblika okvirja, ki omogoča premikanje in spreminjanje velikosti kometa.



Slika 4.3: Okno z vsemi informacijami o kometu

4.5 Urejanje označenih kometov

Z desnim klikom na okvir kometa odpremo meni, ki omogoča nekatere dodatne funkcije (slika 4.2, levo).

Gumb "Inspect" odpre okno z vsemi znanimi informacijami o kometu (slika 4.3). Prikazana je slika kometa, njegov profil in vsi izračunani parametri, poleg tega pa tudi podatki o lokaciji kometa na sliki in njegova velikost v slikovnih točkah.

Možnost "Delete" izbriše označeni komet, izbris pa je možen tudi s priti-

skom na tipko "Delete" na tipkovnici. Ko označeni komet izbrisemo s slike, je izbrisan tudi na seznamu obravnavanih kometov.

S klikom na "Move / Resize" se izgled okvirja izbranega kometa spremeni v obliko, ki omogoča premikanje in spreminjanje velikosti kometa (slika 4.2, desno). Ko uporabnik konča s prirejanjem pozicije in velikosti, s klikom na prazno površino slike okvir kometa povrne v normalno obliko.

Vrstica "Background" nam ponuja dve izbiri: "On bottom" in "On top". Gre za položaj okvirja za ozadje, ki se lahko nahaja nad ali pod okvirjem kometa. Ta je ob označevanju kometa izbran samodejno, vseeno pa ima uporabnik možnost ročne spremembe.

4.6 Izračun in prikaz rezultatov

Za vsak označen komet so po navodilih iz razdelka 2.3.2 samodejno izračunani vsi potrebni parametri za ocenjevanje. Označeni kometi so prikazani v seznamu obravnavanih kometov. Vrednosti parametrov so odvisne od pozicije meje med glavo in repom kometa, zato je komet ponovno analiziran ob vsakem premiku meje. Ponoven izračun parametrov je potreben ob spremembi velikosti ali pozicije kometa in ob spremembi pozicije okvirja za ozadje.

4.7 Opozorila

V razdelku 2.3.3 smo omenili nekatere neželene pojave, ki običajno pomenijo, da komet ni primeren za ocenjevanje. V primeru, da obravnavamo komet s takšnimi lastnostmi, mu aplikacija dodeli status kometa z opozorili. Kometi z opozorili so obarvani rdeče, z desnim klikom na njihov okvir pa je v meniju prikazana tudi opcija "Warnings", ki odpre okno s podrobnostmi o opozorilih kometa.

V oknu s podrobnostmi so prikazana vsa možna opozorila. Pri opozorilih nasičenja (underexposure, overexposure) je prikazano tudi število slikovnih točk s previsoko ali prenizko vrednostjo ter odstotek površine celotnega ko-

```
<?xml version="1.0" encoding="utf-8"?>
  <Files>
    <File>
      <Path>C:\Slike\Snap-0009.tif</Path>
      <Comets>
        <Comet X="287" Y="264" Width="284" Height="178"
          DividerLocation="90" Name="Snap-0009_0"
          Number="1" BackgroundBottom="False" />
        <Comet X="1067" Y="151" Width="260" Height="164"
          DividerLocation="68" Name="Snap-0009_1"
          Number="2" BackgroundBottom="False" />
        <Comet X="843" Y="617" Width="420" Height="301"
          DividerLocation="156" Name="Snap-0009_2"
          Number="3" BackgroundBottom="False" />
      </Comets>
    </File>
  </Files>
```

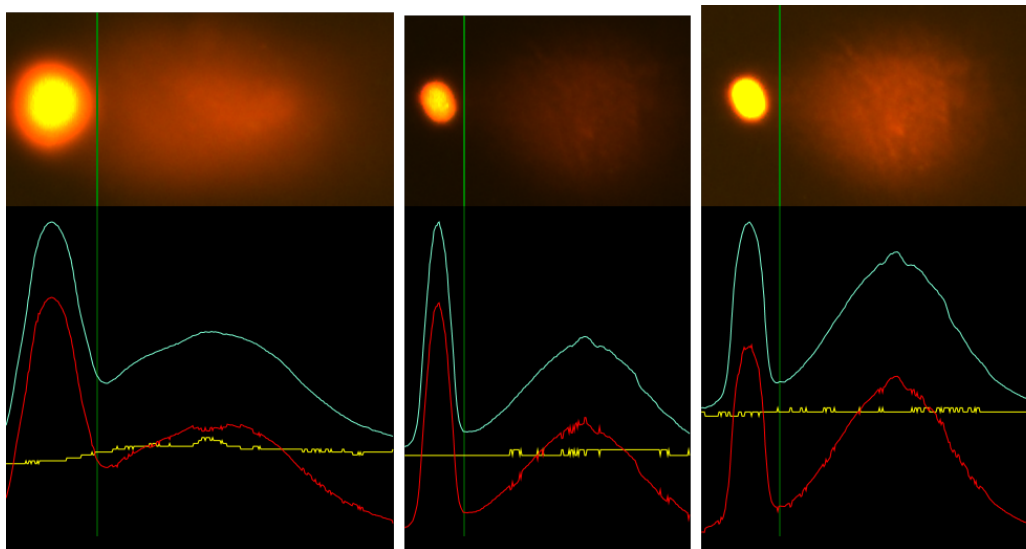
Slika 4.4: Primer datoteke XML s podatki o delu

meta, ki ga te točke zasedajo. Poleg opozoril o neželenih pojavih, omenjenih v razdelku 2.3.3, je prikazano tudi opozorilo o prekrivanju kometov - če imata dva označena kometi skupne točke (torej se vsaj delno prekrivata), za ocenjevanje nista primerna.

4.8 Izvoz in shranjevanje rezultatov

Program je namenjen analizi večjega števila slik (z večimi kometi), zato je bila ena izmed zahtev pri razvoju tudi možnost shranjevanja in kasnejšega nalaganja dela. Shranjevanje rezultatov je mogoče z izbiro *"File → Save work"*, nalaganje pa *"File → Load work"*.

Ob shranjevanju rezultatov se vsi potrebni podatki o naloženih datotekah in obravnavanih kometih zapišejo v datoteko XML. Za vsako sliko se zapiše njena lokacija na disku in seznam kometov, ki so na tej sliki označeni. Za vsak komet pa se hranijo njegove koordinate x in y , višina, širina, ime, zaporedno število, lokacija meje med glavo in repom ter lokacija okvirja ozadja. Primer



Slika 4.5: Rezultat izvoza slik in njihovih profilov

takšnega dokumenta XML je prikazan na sliki 4.4. Vrednosti parametrov za ocenjevanje se v to datoteko ne zapisujejo, ob nalaganju se za vsak komet izračunajo ponovno.

Implementirali pa smo tudi možnost izvoza rezultatov in sicer v dveh oblikah. V prvi obliki gre za združevanje slik obravnavanih kometov in njihovih profilov, sprožimo jo s klikom na gumb "Save image". Program zgenerira sliko v formatu PNG, v kateri so vodoravno združeni profili vseh obravnavanih kometov (slika 4.5). V drugi obliki pa izračunane parametre izvozimo v CSV tabelo, ki jo je mogoče pregledovati v Excell in podobnih aplikacijah. Izvoz je mogoč z gumbom "Export data".

4.9 Nastavitve

Delovanje programa v veliki meri določajo nastavitve. Spreminjamo jih lahko z izbiro "Settings → Open settings", ki nam odpre novo okno z nastavitvami. Okno je razdeljeno na tri zavihke:

- General settings - splošne nastavitve, kjer izbiramo parametre za ocenjevanje kometov in povečavo mikroskopa, pod katero so bile zajete slike. Tu se nahaja tudi opcija izklopa hitrih nastavitvev.
- Calibration - kalibracija orodja, kjer določamo, koliko mikrometrov v fizičnem svetu predstavlja vsaka slikovna točka na slikah, posnetih pod mikroskopom. Pravilna kalibracija je bistvenega pomena za pravilen izračun parametrov komete (dolžina repa, tail extent moment, olive tail moment).
- Automatic detection - nastavitve samodejnega označevanja kometov. Tu lahko izbiramo uporabljen algoritem za upravljanje slike in po potrebi določamo njegove parametre. Na voljo so tudi nastavitve omejevanja velikosti in kontrasta kometov.

Možno je tudi shranjevanje in nalaganje nastavitvev, z izbiro "*Settings* → *Save settings*" in "*Settings* → *Load settings*". Za realizacijo te funkcije smo uporabili podoben pristop kot pri shranjevanju in nalaganju rezultatov. Podatki o nastavitvah se ob shranjevanju zapišejo v XML datoteko, pri nalaganju pa se iz nje preberejo in naložijo v pomnilnik.

Ob zapiranju programa se trenutne nastavitve samodejno zapišejo v datoteko "*lastusage.xml*", ob zagonu pa se iz nje samodejno naložijo. Tako smo se izognili potrebi po nalaganju nastavitvev ob vsakem zagonu programa.

Poglavje 5

Zaključek in nadaljnje delo

Opisali smo kometni test in program za ocenjevanje poškodb DNK na njegovi podlagi. Pripravili smo enostaven uporabniški vmesnik, ki za uporabo ne zahteva obširnega predhodnega znanja, pri čemer smo uporabili moderne tehnologije za razvoj programske opreme na platformi Windows. Razvili smo program, ki omogoča enostavno ročno obravnavanje kometov kot tudi avtomatizirano zaznavanje, poleg tega pa rezultate analize podaja v jasni in pregledni obliki. Podatke, pridobljene z analizo, je mogoče shraniti za kasnejšo obravnavo ali izvoziti v več različnih formatov. Razvita programska oprema zadovolji vse glavne zahteve in je primerljiva z drugimi orodji za delo na tem področju.

Program bi največ nadaljnjega dela in razvoja potreboval na področju avtomatiziranega označevanja kometov z izboljšanjem natančnosti, kot tudi optimizacijo hitrosti zaznavanja. Trenutno implementirane metode so sicer v veliko pomoč pri delu, vseeno pa je področje računalniškega zaznavanja zelo kompleksno in ponuja vrsto drugih rešitev in izboljšav.

Možnosti za nadaljnji razvoj so tudi v vpeljevanju dodatnih zmožnosti programa, ki so na voljo v nekaterih komercialnih orodjih (npr. *single click* označevanje), kot tudi v razširjanju nabora nastavitev, vpeljevanju funkcije povečevanja slik, povezovanje s podatkovnimi bazami ipd.

Slike

2.1	Primer kometa	4
2.2	Ozadje kometa	7
2.3	Primer profila kometa	8
2.4	Neželeni pojavi	9
2.5	Uporabniški vmesnik CometScore Pro	11
2.6	Uporabniški vmesnik CASP	12
3.1	Uporabniški vmesnik Microsoft Visual Studio	14
3.2	Primer kode v jeziku XAML	17
4.1	Uporabniški vmesnik programa	20
4.2	Urejanje kometov	24
4.3	Okno z vsemi informacijami o kometu	24
4.4	Primer datoteke XML s podatki o delu	26
4.5	Rezultat izvoza slik in njihovih profilov	27

Literatura

- [1] B. Likar, "Biomedicinska slikovna informatika in diagnostika", Ljubljana: Založba FE in FRI, 2008, str. 173-174
- [2] M. MacDonald, "Pro WPF 4.5 in C#", ZDA: Apress, 2012
- [3] J. Liberty, "Programming C#, 4th edition", ZDA: O'Reilly Media, 2005
- [4] Vilhar B (2004). "Help! There is a comet in my computer!", http://botanika.biologija.org/exp/comet/comet_guide01.pdf
- [5] Microsoft C#. Dostopno na: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [6] Microsoft Visual Studio. Dostopno na: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- [7] Microsoft Windows Presentation Foundation. Dostopno na: https://en.wikipedia.org/wiki/Microsoft_WPF
- [8] Otsu's method. Dostopno na: https://en.wikipedia.org/wiki/Otsu%27s_method
- [9] XAML. Dostopno na: https://en.wikipedia.org/wiki/Extensible_Application_Markup_Language
- [10] XML. Dostopno na: <https://en.wikipedia.org/wiki/XML>

- [11] Microsoft .NET Framework. Dostopno na:
https://en.wikipedia.org/wiki/.NET_Framework