

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Porovnávání křivek glykovaného hemoglobinu**

## **Prohlášení**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. 5. 2016

Jakub Hain

## **Abstract**

This bachelor thesis deals with development of multiplatform application for evaluating splines of glycated hemoglobin. The developed application should be fast and simple to use. The thesis will describe what glycated hemoglobin is and the problems connected with it. The problem of diabetes is mentioned and given a brief description of a few types included. Then it gives a description of mathematical methods, which are used to evaluate curves of glycated hemoglobin. Framework *Qt* will be used for the realization of the graphical user interface specifically *Qt Widgets*. Then it describes the problem as well as solving of the problem, finally the time consumption tests are mentioned

## **Abstrakt**

Tato práce se zabývá vývojem aplikace pro vyhodnocování křivek hodnot glykovaného hemoglobinu. Vyvíjená aplikace by měla být rychlá a jednoduchá na použití. Tato práce vás na začátku seznámí s glykovaným hemoglobinem a problémy s ním spojené. Je zde popsán problém diabetu a stručný popis jednotlivých typů. Poté jsou v práci uvedeny metody, které budou sloužit k ohodnocení pacientů. Pro vývoj grafického uživatelského rozhraní bude použit framework *Qt*, specificky část *Qt Widgets*. Dále práce se zabývá popisem aplikace a poté samotnou realizací problému. Na závěr je uvedena časová náročnost programu.

## Obsah

|       |   |    |
|-------|---|----|
| 1     | Úvod.....   | 1  |
| 2     | Diabetes.....                                       | 2  |
| 2.1   | Diabetes 1. typu .....                              | 2  |
| 2.2   | Diabetes 2. typu .....                              | 3  |
| 3     | Glykovaný hemoglobin.....                           | 4  |
| 3.1   | Potřeba aplikace.....                               | 5  |
| 4     | Metody pro vyhodnocení pacientů.....                | 6  |
| 4.1   | Metody pro výpočet konvergence .....                | 6  |
| 4.1.1 | Lineární metoda nejmenších čtverců.....             | 6  |
| 4.1.2 | Medián diferencí hodnot .....                       | 7  |
| 4.1.3 | $C*e^{(K*x)}$ .....                                 | 7  |
| 4.2   | Metody pro výpočet variability .....                | 8  |
| 4.2.1 | Směrodatná odchylka .....                           | 8  |
| 4.2.2 | Směrodatná odchylka diferencí hodnot s ofsetem..... | 8  |
| 4.2.3 | Mezikvartilové rozpětí .....                        | 9  |
| 5     | Analýza .....                                       | 10 |
| 5.1   | C++ .....   | 10 |
| 5.2   | Qt .....  | 10 |
| 5.2.1 | Signal-Slot.....                                    | 11 |
| 5.3   | Vstupní soubor.....                                 | 11 |
| 5.4   | Výběr knihovny pro načtení xlsx.....                | 12 |
| 5.5   | Vzhled aplikace .....                               | 12 |
| 5.5.1 | Multiple document interface .....                   | 12 |
| 5.6   | Graf hodnot HbA1c .....                             | 13 |
| 5.7   | Graf rozdělení do skupin .....                      | 13 |
| 5.7.1 | Barvy grafu.....                                    | 13 |

|       |  |    |
|-------|--|----|
| 5.8   | Model/View architektura .....            | 14 |
| 5.9   | Rozdělení do skupin .....                | 16 |
| 6     | Řešení.....                              | 17 |
| 6.1   | Použité technologie.....                 | 17 |
| 6.2   | Detail implementace .....                | 17 |
| 6.2.1 | Zpracování vstupu z xlsx.....            | 17 |
| 6.2.2 | Výstup ve formátu xlsx .....             | 18 |
| 6.2.3 | MainWindow.....                          | 19 |
| 6.2.4 | Grafy.....                               | 19 |
| 6.2.5 | Nastavení.....                           | 20 |
| 6.2.6 | Tabulky.....                             | 22 |
| 6.2.7 | Export grafů do SVG.....                 | 24 |
| 6.2.8 | Překlad aplikace do češtiny .....        | 25 |
| 6.2.9 | Výpočty metod .....                      | 27 |
| 6.3   | Výkonnost aplikace .....                 | 27 |
| 6.3.1 | Načítání vstupu.....                     | 28 |
| 6.3.2 | Export do SVG .....                      | 28 |
| 6.3.3 | Ukládání xlsx souboru.....               | 29 |
| 6.4   | Vytvoření instalačního souboru.....      | 29 |
| 7     | Analýza hodnot HbA1c.....                | 31 |
| 8     | Závěr .....                              | 34 |
|       | Seznam zkratk .....                      | 35 |
|       | Zdroje .....                             | 36 |
|       | Příloha A: Uživatelský manuál .....      | 37 |
|       | Příloha B: Zjednodušený uml diagram..... | 42 |

# 1 Úvod

Diabetes je jedním z nejrozšířenějších onemocnění a počet lidí s touto nemocí stále roste. Z tohoto důvodu je důležité neustále se snažit zlepšovat způsoby léčby. Diabetes nelze úplně vyléčit, ale lze jeho následky omezit.

Pro lidi s diabetem je důležité, aby jim byla kontrolována hodnota glykovaného hemoglobinu (HbA1c), test HbA1c je důležitý, protože udává dlouhodobý přehled o tom, jaká je průměrná hodnota HbA1c.

Při léčbě diabetu je snaha, aby tato hodnota klesla co nejbližší k normální hranici nebo k hranici, kterou určí lékař. Z tohoto důvodu je důležité testovat, zda dlouhodobě pacientova hodnota HbA1c klesá. Pacientova hodnota HbA1c by také neměla být vysoce variabilní, tedy neměla by se náhle zvyšovat nebo snižovat. Z tohoto důvodu se testuje variabilita hodnot HbA1c pacienta.

Tato práce se tedy bude touto problematikou zabývat a cílem je vývoj aplikace, která bude schopna vyhodnotit data jednotlivých pacientů a bude je schopna přehledně zobrazit.

V následujících kapitolách bude nejprve představeno co je diabetes a jeho nejrozšířenější typy. Dále bude stručně popsáno co je glykovaný hemoglobin. Poté bude provedena analýza problému. Poslední část práce popisuje samotnou realizaci problému a vyhodnocení výsledků.

## 2 Diabetes

Podle statistik lidí s diabetem postupně přibývá, toto je možná způsobeno zlepšením diagnostiky diabetu a celkovému zlepšení lékařské péče.

|      | DM 2. typu | DM 1. typu | Sekundární diabetes | DM celkem |
|------|------------|------------|---------------------|-----------|
| 2012 | 772 585    | 56 514     | 12 128              | 841 227   |
| 2011 | 758 719    | 55 542     | 11 121              | 825 382   |
| 2010 | 739 859    | 55 811     | 10 560              | 806 230   |
| 2009 | 717 365    | 55 414     | 10 542              | 783 321   |
| 2008 | 708 847    | 54 474     | 10 240              | 773 561   |
| 2007 | 692 074    | 52 813     | 10 074              | 755 000   |
| 2006 | 686 159    | 51 070     | 11 299              | 749 000   |
| 2000 | 599 868    | 46 446     | 8 504               | 654 164   |
| 1997 | 555 883    | 39 020     | 5 402               | 600 306   |
| 1975 | –          | –          | –                   | 234 071   |

*Obr. 1 Statistika lidí s diabetem, Zdroj: ÚZIS*

Diabetes vzniká nedostatkem nebo nedostatečným využitím inzulínu. Inzulín je produkován ve slinivce břišní a umožňuje buňkám využít glukózu, pro které je glukóza hlavním zdrojem energie. Pokud je inzulínu nedostatek nebo není dostatečně využit, dochází k hromadění glukózy (hyperglykémii). Pokud dojde k nedostatku množství potravin nebo je nedostatečný příjem potravin oproti tělesné aktivitě, může dojít ke snížení glukózy v krvi (hypoglykémii). Nejrozšířenějšími druhy diabetu je diabetes 1. a 2. typu. Ostatní druhy se vyskytují v menším množství, jako gestační diabetes a další specifické typy.

### 2.1 Diabetes 1. typu

Diabetes 1. typu obvykle začíná v dětství nebo během dospívání. Tímto typem onemocnění trpí 7% diabetiků v České republice. Pro tento typ diabetu je charakteristická úplná absence inzulínu v těle. Tento typ diabetu se projeví, až když je zničeno přibližně 90% buněk produkujících inzulín. Proto lidé s diabetem 1. typu musí inzulín do těla dodávat injekčně.

## 2.2 Diabetes 2. typu

Diabetem 2. typu u nás trpí 92% diabetiků. Tento typ postihuje spíše starší osoby a osoby s nadváhou. Genetické dispozice také přispívají ke vzniku tohoto onemocnění. Oproti 1. typu mají pacienti s diabetem 2. typu normální hladinu inzulínu nebo dokonce nadbytek.

S diabetem 2. typu také souvisí inzulínová rezistence. Inzulínová rezistence je stav, při kterém lidské tělo nedokáže efektivně využít produkovaného inzulínu. Při tomto onemocnění dochází k hromadění inzulínu v krvi, místo jeho absorbování tělesnými buňkami. Inzulínovou rezistenci je možné léčit nejprve změnou životního stylu, dietním opatřením a zvýšením fyzické aktivity.



### 3 Glykovaný hemoglobin

Termín *HbA1c* značí glykovaný hemoglobin, který vzniká při spojení hemoglobinu a glukózy v krvi.

Měřením *HbA1c* je možné získat lepší přehled o stavu krevního cukru během uplynulých týdnů. Tato doba je rovna době života červených krvinek. Měření *HbA1c* zvýšilo úspěšnost nálezu diabetu. Pro lidi s diabetem je důležité sledovat tuto hodnotu, protože čím vyšší hodnota, tím vyšší šance na vyvinutí problémů spojených s diabetem.

| Hranice        | IFCC mmol/mol    | DCCT%       |
|----------------|------------------|-------------|
| Normální       | < 42 mmol/mol    | < 6.0%      |
| Riziko diabetu | 42 - 47 mmol/mol | 6.0% - 6.4% |
| Diabetes       | > 47 mmol/mol    | > 6.4%      |

Tab. 1 Hraniční hodnoty diabetu

Pro diabetiky je při léčbě dobrým postupem přiblížit se hodnotě *HbA1c* 6,5% nebo 48mmol/mol. Tato hodnota je pouze orientační a záleží na lékaři, aby určil k jaké hodnotě *HbA1c* by měl člověk s diabetem směřovat.

Tento test je běžně prováděn při kontrole krve. I když riziková hranice je 42 mmol/mol, tak je dobré upozornit pacienty už při 39 mmol/mol.

$$\begin{aligned} NGSP &= (0.09148 * IFCC) + 2.152 \\ IFCC &= (10.93 * NGSP) - 23.5 \end{aligned} \quad (3-1)$$

Glykovaný hemoglobin je možný uvádět dle různých standardů, například IFCC nebo NGSP, převod mezi těmito standardy viz vzorce 3-1. V České republice se používá standard IFCC, do 31. 12. 2011 byly hodnoty uváděny v procentech. V dnešní době se již užívají jednotky mmol/mol. Přepočítání mezi těmito způsoby je jednoduchý procentuální výsledek stačí vynásobit deseti a dostaneme mmol/mol hodnotu, pro zjištění procentuální hodnoty je postup obrácen.

Je možné testovat i úroveň glukózy v krvi. Tento test poskytne současnou úroveň, ale je snadno ovlivnitelný. Pokud si testovaná osoba dá večer sladké jídlo, tak tento test ukáže vyšší hodnotu než normálně. Tyto hodnoty měří například domácí glukometr. Test *HbA1c* snadno ovlivnit nelze, to ale neznamená, že by byl úplně odolný. Například pokud by člověk často konzumoval alkohol ve větším množství nebo

měl problémy s ledvinami, je možné, že tato hodnota bude vyšší než normálně. Stejně tak existují způsoby, které ovlivní výsledek na druhou stranu.

### **3.1 Potřeba aplikace**

Tato aplikace je vyvíjena, protože při dlouhodobé léčbě pacientů může docházet k různým změnám, jak u pacienta, tak ve způsobu léčby. Z tohoto důvodu je dobré, aby lékař měl přístup k aplikaci, která data zpracuje a bude je schopna přehledně zobrazit.

Aplikace bude testovat pouze konvergenci a variabilitu hodnot HbA1c pacienta, ale bude velmi jednoduchá na rozšíření pro další metody a způsoby hodnocení.

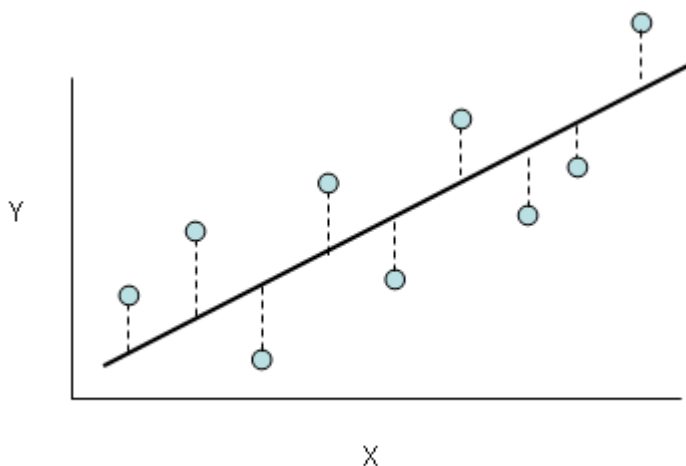
## 4 Metody pro vyhodnocení pacientů

### 4.1 Metody pro výpočet konvergence

Pro pacienty s diabetem je důležité, aby hodnota *HbA1c* nestoupala, ale naopak se snižovala někam k hranici 48 mmol/mol. Při zvyšování hodnoty je pravděpodobné, že zvolená léčba je neefektivní a je třeba ji změnit.

#### 4.1.1 Lineární metoda nejmenších čtverců

Metoda nejmenších čtverců je jedním ze způsobů řešení přeuročených soustav rovnic, také se tato metoda používá v regresní analýze. Tato metoda se dá použít na aproximace libovolným polynomem, ale pro potřeby této práce bude použita aproximace přímkou, tedy  $f(x) = k_1x + k_0$ , kde  $k_1$  bude určovat, zda metoda konverguje k přijatelné hodnotě *HbA1c*. Pro tuto práci je vhodné, aby přímka byla ostře klesající. To znamená, že pro tento případ jsou přijatelné všechny záporné  $k_1$ .



Obr. 2 Výsledná přímka metody nejmenších čtverců

Tato metoda se snaží minimalizovat sumu obsahů čtverců. Pokud by body aproximované křivky byly  $y'$  a naměřené hodnoty  $y$ . Pak se snažíme minimalizovat sumu, kterou získáme vytvořením čtverců mezi jednotlivými body  $y'$  a  $y$ .

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (4-1)$$

Existuje několik metod řešení metody nejmenších čtverců. Rozhodl jsem se pro řešení pomocí matic. Vytvoříme matici  $A$  a  $Y$  podle vzorce 4-1, kde 1. sloupec matice  $A$  obsahuje pouze číslo 1 a 2. sloupec obsahuje hodnoty  $x$ , pro tuto práci konkrétně čísla  $1 \dots n$ , podle počtu hodnot *HbA1c* pacienta. Matice  $Y$  obsahuje konkrétní hodnoty *HbA1c*. Poté vynásobíme  $Q = A^T A$  a  $B = A^T Y$ , dále už řešíme soustavu  $Qx=B$ . Tuto soustavu vyřešíme pomocí Cramerova pravidla. Pro potřeby semestrální práce není potřeba vypočítat  $k_0$ , takže stačí spočítat determinant  $D$  matice  $Q$  a determinant  $D_2$  matice  $Q$ , ve které nahradíme 2. sloupec maticí  $B$ . Potom  $k_1 = D_2/D$ .

Nevýhodou této metody je velký vliv extrémních hodnot.

#### 4.1.2 Medián diferencí hodnot

Další způsob, který byl využit na zjištění konvergence, je spočítat rozdíly mezi po sobě následujícími hodnotami. Pokud se hodnota snižuje, mělo by být více hodnot záporných než kladných a tedy medián by měl určit, zda hodnoty klesají. U této metody je menší vliv extrémních hodnot, ale zase nelze určit, jak rychle metoda klesá na rozdíl od metody nejmenších čtverců.

Medián je hodnota, jež rozděluje řadu hodnot seřazených vzestupně na dvě poloviny o stejném počtu prvků. V případě sudého počtu prvků se ze dvou nejbližších hodnot udělá aritmetický průměr a toto číslo je považováno za medián.

#### 4.1.3 $C * e^{(K * x)}$

Tato metoda je vypočítána pomocí nejmenších čtverců. Pro vyhodnocení je

$$C = \exp(c), K \equiv k \quad (4-2)$$

$$\begin{bmatrix} \sum_{i=1}^n y_i & \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i^2 y_i^2 \end{bmatrix} \begin{bmatrix} c \\ k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \ln y_i \\ \sum_{i=1}^n x_i y_i \ln y_i \end{bmatrix} \quad (4-3)$$

použit parametr  $k$ . Pro výpočet parametrů  $C$  a  $K$  je využito metody nejmenších čtverců.

Dosazením do vzorce 4-3 a následným vypočtením této soustavy rovnic budou získány hodnoty parametrů  $c$  a  $k$ . Poté rovností ve vzorci 4-2 získáme parametry  $C$  a  $K$ .

Vzhledem k tomu, že hodnoty glykovaného hemoglobinu jsou kladné, tak hodnoty  $k$  by měly být záporné, aby funkce klesala. Parametr  $c$  bude pro tento příklad vždy kladný.

## 4.2 Metody pro výpočet variability

Je dobré, aby hodnoty  $HbA1c$  neměly vysoce rozdílné hodnoty, proto je vhodné testovat, zda je variabilita velká nebo ne.

### 4.2.1 Směrodatná odchylka

Směrodatná odchylka je ve statistice často používána pro určení statistické disperze. To znamená, že určuje, jak moc se hodnoty od sebe liší.

$$\sigma(x) = \sqrt{\frac{1}{N}(x - \bar{x})^2} \quad (4-4)$$

Pro výpočet směrodatné odchylky (vzorec 4-4) je nejprve nutné určit aritmetický průměr. Poté se vypočítá rozdíl jednotlivých hodnot od aritmetického průměru, z těchto hodnot se vypočte druhá mocnina a hodnoty se sečtou. Poté se tato suma vydělí počtem hodnot  $N$ . Tato hodnota určuje rozptyl hodnot. Standardní odchylka je rovna odmocnině z rozptylu.

U standardní odchylky platí, že čím více se hodnota blíží k 0, tím méně se od sebe hodnoty liší.

### 4.2.2 Směrodatná odchylka diferencí hodnot s offsetem

Touto metodou je umožněno vynechat určitý počet hodnot  $HbA1c$  od počátku měření. Tato metoda je vhodná pro zjištění vývoje v průběhu léčby např. pokud zjišťuji, zda je po změně způsobu léčby léčba stále úspěšná.

Na rozdíl od metody v oddílu 5.1 tato metoda počítá se diferencemi jednotlivých hodnot.

### 4.2.3 Mezikvartilové rozpětí

Tato statistická metoda slouží k určení rozpětí mezi 1. a 3. kvartilem. Při výpočtu mezikvartilového rozpětí (vzorec 4-5) musíme nejprve seřadit hodnoty podle velikostí vzestupně. Poté se řada musí rozdělit na 2 stejné poloviny, při lichém počtu prvků se středový člen vynechává a použijí se prvky kolem. A následně se určí medián obou polovin. Medián spodní poloviny je  $Q1$  a medián horní poloviny je  $Q3$ . Odečtením  $Q1$  od  $Q3$  pak získáme mezikvartilové rozpětí.

$$IRQ = Q3 - Q1 \quad (4-5)$$

Tímto způsobem získáme rozpětí mezi 25% a 75% percentilem. Tím odstraníme nejméně a nejčastěji se vyskytující prvky.

## 5 Analýza

### 5.1 C++

Program byl napsán v programovacím jazyce C++. Tento jazyk vznikl již na počátku 80. let minulého století. Je to jazyk, který podporuje několik programovacích paradigmat. Podporuje procedurální, objektově orientované nebo generické programování. Nejedná se tedy o čistě objektový jazyk.

Jazyk C++ vyvíjel Bjarne Stroustrup v Bellových laboratořích od roku 1979, jako rozšíření jazyka c. V současné době je platný ISO standard C++14 nebo ISO/IEC 14882:2014, který byl vydán v roce 2014, další revize je plánována na rok 2017.

### 5.2 Qt

K vytvoření uživatelského rozhraní bude využit framework *Qt*<sup>1</sup>. Tento framework je jedním z nejrozšířenějších v oblasti vytváření GUI (Graphical User Interface) a je to framework s otevřeným kódem (OSS - *Open-Source Software*). Je napsán v jazyce C++, ale existují i odnože např. *QtRuby*, *PyQt*, *PySide* atd. *Qt* je dnes vyvíjen společností *Qt Company*, která je dceřinou společností firmy *Digia*, které *Qt* patří od roku 2011 po převzetí od společnosti *Nokia*. Velkou výhodou tohoto frameworku je jeho multiplatformnost. Od verze 5.5. *Qt* umožňuje 2 rozdílné přístupy k vytváření grafického rozhraní *Qt Quick* a *Qt Widget*.

*Qt Widget* je sbírka komponent potřebných pro vytvoření GUI. Tyto komponenty jsou základními stavebními bloky aplikace, je samozřejmě možné vytvořit si widgety nové nebo upravit již existující. Změna vzhledu widgetů je možná pomocí *QStyle*, který využívá *css* stylů. Tento způsob vytváření grafického rozhraní je hodně podobný například *JavaFX*. Třída *Qt Widget* umožňuje základní schopnosti zobrazování na obrazovku a zpracování uživatelských vstupů.

*Qt Quick* je novější než *Qt Widget*. Využívá jazyka *QML* (*Qt Modeling Language*), který je založený na *javascriptu*, k popisu GUI. *QML* bylo vyvinuto pro snadnou tvorbu dynamického GUI. Je tedy velmi vhodný pro aplikace na tablety a

---

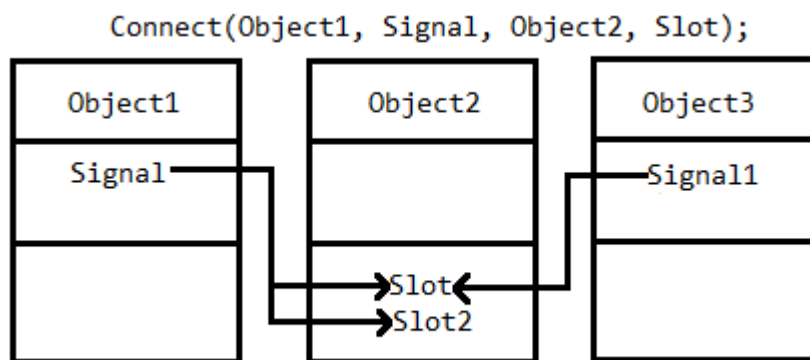
<sup>1</sup> <http://www.qt.io/>

mobilní telefony. Stejně jako *Qt Widget* obsahuje sbírku komponent, a tedy nic nebrání tomu vyvíjet i aplikace pro desktop.

### 5.2.1 Signal-Slot

Pro komunikaci mezi objekty je použito signálů a slotů. Toto je jedním z nejdůležitějších mechanismů *Qt*.

Při vytváření GUI, když provedeme nějakou změnu (např. stisknutí tlačítka) obvykle chceme, aby změna byla viditelná. Toho dosáhneme pomocí signálů a slotů. Při každé akci je volán určitý signál. Slot je poté funkce volaná při určitém signálu. Jednotlivé komponenty mají předdefinované sloty, ale často je potřeba přidat vlastní sloty pro námi požadované funkce. Pro spojení signálu a slotu je využito funkce `connect`.



*Obr. 3 Ukázka funkce signálů a slotů, příkaz pro spojení*

Ke každému signálu je možné přiřadit několik slotů, je i možné spojit signál s jiným signálem. Stejně tak je možné k jednomu slotu přiřadit několik signálů. Sloty je možné volat i jako normální funkce *C++*.

### 5.3 Vstupní soubor

Aplikace bude schopna zpracovat vstupní soubor ve formátu `xlsx`. Tento soubor bude obsahovat id pacientů a jejich naměřené hodnoty `HbA1c`. První řádek souboru obsahuje popis jednotlivých sloupců. Dále už jsou jednotlivé řádky s daty ve formátu `1`. sloupec obsahuje jméno nebo id pacienta, další sloupce už obsahují hodnoty `HbA1c`. Nejstarší hodnoty se nacházejí vlevo a s postupem vpravo jsou hodnoty novější. Hodnoty jsou tedy seřazeny chronologicky od nejstarších po nejnovější.



## 5.4 Výběr knihovny pro načtení xlsx

Na výběr bylo několik knihoven, ale některé nespĺňovaly všechny požadavky, tedy čtení a zápis do xlsx a možnost vytvořit graf. Rozhodoval jsem se tedy mezi knihovnami *QtXlsx*<sup>2</sup> a *SimpleXlsxWriter*<sup>3</sup> nakonec jsem zvolil *QtXlsx*, protože je možné ho přidat jako addon do *Qt*. *QtXlsx* je Open-Source projekt a podporuje veškeré platformy, na kterých je možné použít *Qt*.

## 5.5 Vzhled aplikace

Při vytváření aplikací je velmi důležitou částí vytvoření uživatelsky přívětivého a na použití jednoduchého uživatelského rozhraní. Protože je to, to s čím se uživatel setká, má uživatelské rozhraní velký vliv na celkový dojem z aplikace.

Aplikace se bude skládat ze tří hlavních částí. Pravý sloupec bude obsahovat tabulku zobrazující všechny pacienty a jejich hodnocení vybranou metodou. Tato tabulka bude read-only a bude mít povolenou možnost výběru více položek, z této tabulky pak bude možné vybrat pacienty, ze kterých bude vykreslen graf.

Dolní část okna bude zaplňovat tabulka, která bude obsahovat všechny pacienty a k nim všechny vypočtené hodnoty konvergence a rozptylu.

Hlavní část okna zabere mdi (*Multiple document interace*) oblast, ve které budou zobrazeny grafy zvolených pacientů. Tyto grafy pak půjdou dále exportovat do SVG.

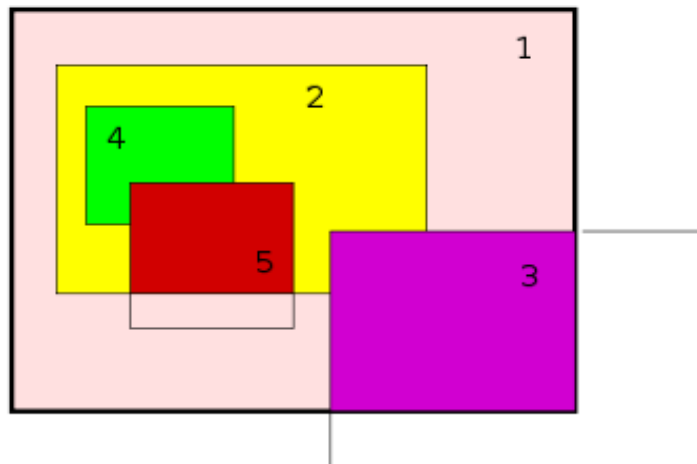
### 5.5.1 Multiple document interface

Je grafické rozhraní, ve kterém několik oken má pouze 1 otcovské okno. Tato metoda vytváření grafického rozhraní tedy umožňuje, aby více oken bylo zapouzdřeno v jiném okně. V základu tento způsob zobrazení připomíná ve svém chování klasickou plochu.

---

<sup>2</sup> <http://qtxlsx.debao.me/>

<sup>3</sup> <http://simplexlsx.sourceforge.net/>



Obr. 4 Ukázka mdi okna

## 5.6 Graf hodnot HbA1c

Každá linka v grafu bude reprezentovat jednoho pacienta. V grafu bude nutné ošetřit, aby jednotlivé linky byly dobře barevně odlišeny a graf byl dobře čitelný při rozumném počtu linek. Osa x bude reprezentovat jednotlivé naměřené údaje, zatímco osa y bude reprezentovat hodnotu *HbA1c* pacienta. Osy a linky budou reprezentovány, jako *QpainterPath* frameworku *Qt*. Graf bude možné zmenšit, či zvětšit podle velikosti okna.

## 5.7 Graf rozdělení do skupin

Tento graf bude zobrazovat data vyhodnocených metod. A podle zvolené metody konvergence a variability poté zobrazí vybrané pacienty ve formě grafu. Na ose x je zobrazena konvergence a osa y zobrazuje variabilitu. Každý kvadrant tohoto grafu reprezentuje jednu ze skupin. S tím, že u variability i konvergence platí, že čím nižší hodnota tím příznivější. Z tohoto důvodu má 3. kvadrant nejlepší hodnocení.

### 5.7.1 Barvy grafu

Existuje mnoho způsobů, jak vytvořit různé barvy, rozhodoval jsem se mezi využitím algoritmu na vytvoření barev nebo vytvořením statického seznamu, který bude poté určovat barvy jednotlivých linek.

Jednou z prvních možností je generovat barvy náhodně, tato metoda vrací zajímavé výsledky, ale není dostatečně konzistentní na to, aby se dala použít při vytváření grafu.

Další možností bylo využít vlastností rgb a postupně generovat co nejodlišnější barvy. Nejprve se začne se třemi základními barvami. Po těchto barvách se hodnota vydělí 2 a vytvoří se všechny kombinace barev, kromě šedé. Tato metoda vracela celkem uspokojivé výsledky, ale zabírala čas.



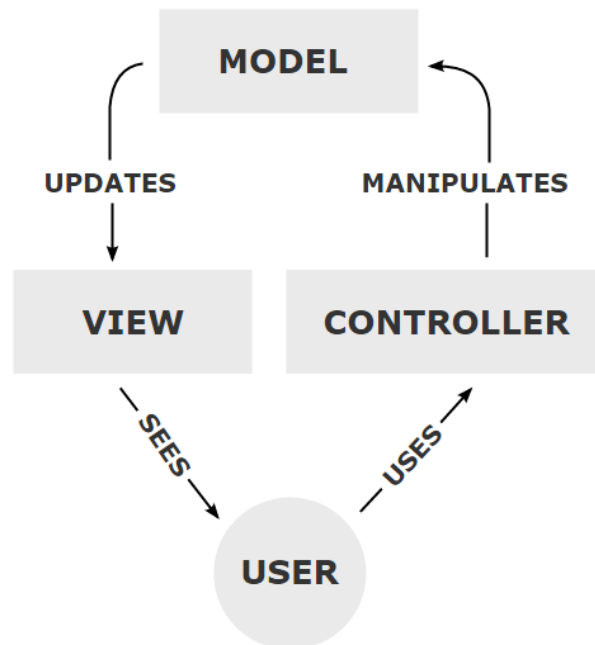
*Obr. 5 64 různých barev*

Nakonec jsem tedy zvolil vhodné barvy a z nich vytvořil statický seznam 63 barev. Pokud by v grafu bylo více než 63 linek, začnou se barvy opakovat, ale při takto vysokém počtu čar se graf stává nepřehledným a barvy čar už nemají moc velký vliv na čitelnost grafu. Proto byla zvolena tato metoda.

## **5.8 Model/View architektura**

*Qt* obsahuje množinu view tříd, které používají model/view architekturu ke správě dat a způsobu jejich zobrazení. Toto rozdělení dává vývojáři větší možnosti upravení prezentace položek.

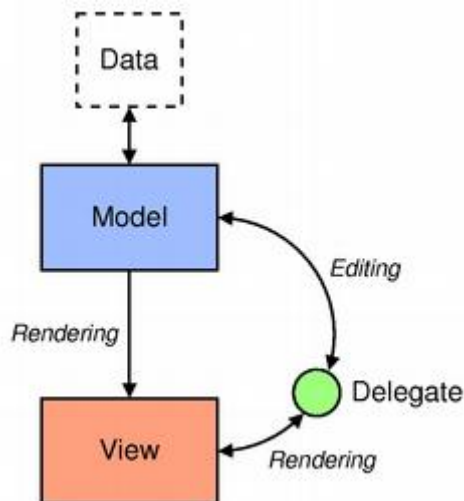
*Model-View-Controller* (MVC) je návrhový vzor, který byl poprvé použit v jazyce *Smalltalk*. Je často používán pro vytváření uživatelských rozhraní. MVC se skládá z 3 částí.



Obr. 6 MVC model

- Model – obsahuje data a výkonný kód pro práci s daty
- View – vizuální stránka aplikace
- Controller – zpracovává vstupy uživatele

*Qt* spojil *view* a *controller* do jednoho objektu, ale nechal model, který spravuje data, proto model/view architektura. Tato metoda stále umožňuje zobrazit stejná data různými způsoby, podle zvoleného view. Ke snazšímu zpracování uživatelského vstupu slouží delegát. Výhodou delegáta je možnost jiného zobrazení položky při editaci.



Obr. 7 Model/view architektura

## 5.9 Rozdělení do skupin

Program bude schopen rozlišit pacienty podle vypočtených metod do celkem 4 skupin. Variabilita má 2 skupiny a konvergence taktéž. Skupiny jsou rozděleny podle mediánu nebo průměru vypočtených hodnot pacientů. Skupina lepších než hraničních hodnot je značena L, horší hodnoty jsou značeny H. Tyto skupiny jsou poté zobrazeny v grafu rozdělení do skupin i v tabulce s rozdělením pacientů do skupin.

Využitím mediánu nebo průměru z vypočtených hodnot ovšem není zamezeno možnosti, že hraniční bod bude ještě ve špatných hodnotách. Při výpočtu konvergence je v programu zajištěno, že u všech hodnot, které jsou záporné, bude křivka hodnot HbA1c pacienta klesat. U variability zase platí, že čím menší je hodnota, tím je křivka stabilnější.

Může zde nastat situace, ve které žádný z pacientů nemá ve výsledku příznivé hodnoty. Z tohoto důvodu byly skupiny pojmenovány horší a lepší, protože nelze podle mediánu nebo průměru určit zda jsou tyto hodnoty již dobré nebo ne.

## 6 Řešení

V této kapitole bude popsáno, jak jsou jednotlivé části implementovány a jaké třídy a nástroje jsou využity.

### 6.1 Použité technologie

Pro vytvoření uživatelského rozhraní bude použit framework *Qt* verze 5.5.1, hlavně jeho součást *QT Widget*, jak je popsáno v kapitole 6.2.

Z počátku tvorby aplikace byl jako *IDE* (Integrated Development Environment) použit *Qt Creator*, později se vývoj aplikace přesunul do *Visual Studio 2013*.

Pro překlad programu z angličtiny do češtiny byl využit *Qt Linguist*.

### 6.2 Detail implementace

Každý *C++* program obsahuje třídu `main(...)`. V této třídě je nastaven překlad a je spuštěna samotná aplikace.

V kódu aplikace je oddělen kód pro výpočty a práci se vstupem od kódu GUI a reakce na vstupy. Tímto je dosaženo snadné možnosti upravení vzhledu.

#### 6.2.1 Zpracování vstupu z *xlsx*

O zpracování vstupu se stará třída `CXLSXreadwrite`. Při načítání *xlsx* souboru je nejprve důležité určit kolik pacientů je v souboru. Toho je dosaženo kontrolou prvního sloupce souboru. Kontrola skončí při prvním prázdném řádku. Poté je vytvořen vektor pacientů a začnou se načítat jednotlivé hodnoty pacientů.

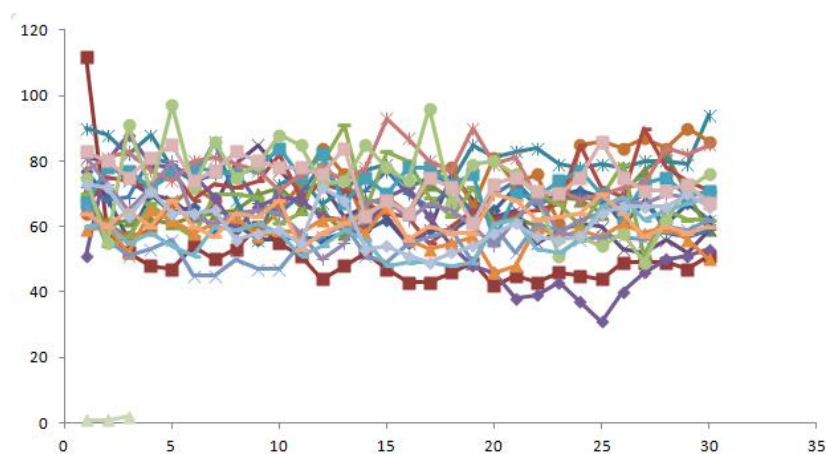
|    | A    | B      | C  | D  | E  | F      | G  | H  | I  | J  | K       |
|----|------|--------|----|----|----|--------|----|----|----|----|---------|
| 1  | PtId | glyHb1 |    |    |    | glyHb5 |    |    |    |    | glyHb10 |
| 2  | 1    | 77     | 69 | 56 | 70 | 64     | 66 | 59 | 61 | 56 | 56      |
| 3  | 2    | 112    | 59 | 52 | 48 | 47     | 54 | 50 | 53 | 60 | 55      |
| 4  | 3    | 71     | 60 | 62 | 62 | 61     | 63 | 64 | 65 | 70 | 72      |
| 5  | 4    | 81     | 80 | 74 | 70 | 68     | 63 | 66 | 79 | 85 | 77      |
| 6  | 5    | 90     | 88 | 82 | 88 | 78     | 74 | 86 | 68 | 56 | 73      |
| 7  | 6    | 64     | 61 | 57 | 59 | 67     | 72 | 68 | 60 | 59 | 65      |
| 8  | 7    | 72     | 68 | 69 | 81 | 79     | 74 | 76 | 77 | 67 | 64      |
| 9  | 8    | 66     | 75 | 74 | 72 | 77     | 68 | 73 | 72 | 74 | 82      |
| 10 | 9    | 70     | 54 | 57 | 67 | 61     | 57 | 70 | 70 | 69 | 61      |

Obr. 8 Část vstupního souboru

Načítání hodnot je provedeno podobně jako zjišťování počtu pacientů. Data se ukládají do vektoru do té doby, dokud pole vstupního souboru není prázdné. Nebo neobsahuje neplatný vstup. Nevýhodou je, že za neplatný vstup je považována 0, protože při převodu z *QVariant* do double je při neplatném vstupu vrácena 0. Toto by ovšem v aplikaci neměl být problém, protože hodnoty *HbA1c* by 0 dosáhnout neměly. Počet načtených hodnot je omezen pouze maximální velikostí vektoru.

## 6.2.2 Výstup ve formátu xlsx

Program je také schopný uložit data do xlsx souboru. Tento soubor poté obsahuje 3 listy. První list obsahuje původní data. Na druhém listu jsou uloženy vypočtené hodnoty všech metod pro všechny pacienty. Poslední list obsahuje graf průběhu hodnot *HbA1c*. Tento graf postrádá legendu, protože použitá knihovna neumožňuje přidání legendy ke grafu. Umožňuje pouze vykreslení různých typů grafu a dat v nich.



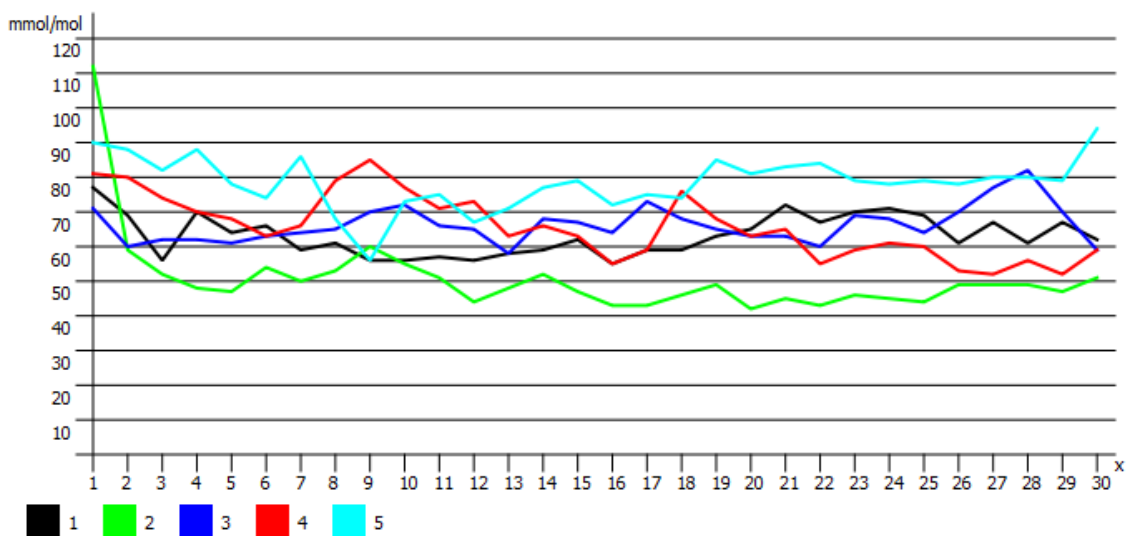
Obr. 9 Ukázka grafu v xlsx souboru

### 6.2.3 MainWindow

Jednou z hlavních částí aplikace je třída `MainWindow`. Tato třída se stará o vytváření GUI, a obsluhu signálů. To znamená, že obsahuje implementaci slotů a veškeré komponenty pro vytvoření GUI.

### 6.2.4 Grafy

Jednou z velmi důležitých funkcí aplikace je vykreslení grafů, tyto funkce obstarávají 2 třídy, pro každý graf 1. Těmito třídami jsou `GraphSubWindow` a `ResultSubWindow`. K vytvoření grafů byly vytvořeny nové widgety. Při vytváření těchto grafů bylo důležité, aby se přizpůsobily velikosti okna. To znamená, že při každé změně okna je nutné přepočítat veškeré body grafu. K dosažení snadného využití tyto třídy dědí `QWidget`. Každý `QWidget` má funkci `painterEvent(...)`, která slouží k vykreslení obsahu. Pro vykreslení grafu je využito třídy `QPainter`. Tato třída umožňuje vykreslení základních útvarů nebo textu.

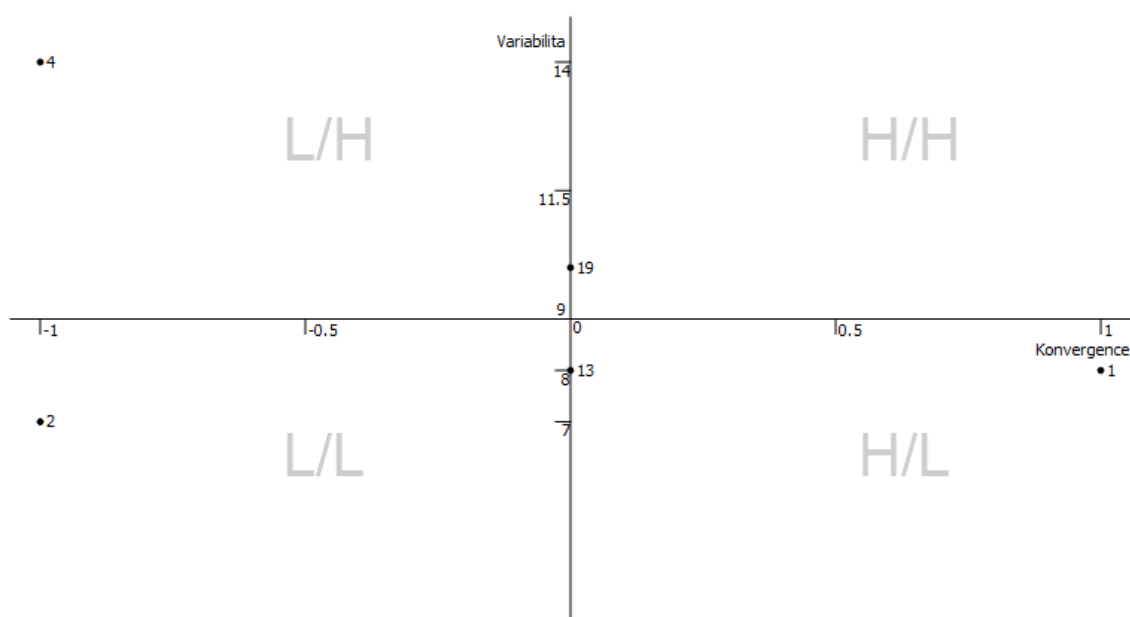


Obr. 10 Graf HbA1c pacientů



GraphSubWindow (obr. 8) slouží k vykreslení průběhu křivek jednotlivých pacientů. U tohoto grafu je důležitá také legenda, která je vykreslena pod grafem. U legendy je problém, že při mnoha pacientech zabírá hodně místa, byla zde možnost omezit počet vykreslených křivek, ale nakonec bylo rozhodnuto zanechat možnost vykreslení libovolného množství křivek.

ResultSubWindow vykreslí, do jakých skupin (viz 6.7) je podle výsledků zvolených metod pacient zařazen. U tohoto grafu jsou téměř všechny souřadnice počítány od středu grafu. Hodnoty variability jsou zobrazeny na ose x a hodnoty konvergence jsou na ose y. Je to dáno tím, že tento graf má posunutý střed os na souřadnice průměru nebo mediánu zvolených metod v nastavení. Vzhledem k tomu, že u všech metod platí, že nižší hodnota je lepší, proto pacienti, kteří jsou zobrazeni ve 3 kvadrantu, mají obě hodnocení lepší než hraniční bod.



Obr. 11 Graf rozdělení pacientů do skupin

## 6.2.5 Nastavení

Pro správnou funkci programu je potřeba konfigurační soubor. Tento soubor obsahuje, jaké metody budou použity při spuštění programu, ofset pro metodu směrodatné odchylky diferencí s ofsetem měsíců (viz 4.2). Konfigurační soubor je realizován textovým souborem s názvem "config.ini". Pokud tento soubor není

```
[Method]
Convergence = 0
Variance = 0
```

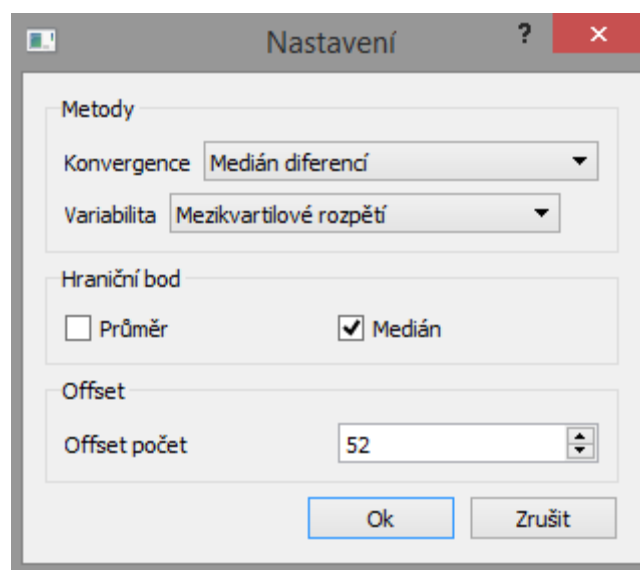
```
[Result]
Median = 1
```

```
[Variable]
Skip = 10
```

```
[Info]
Show = 0
```

Obr. 13 Ukázka konfiguračního souboru

dostupný, pak se při zapnutí programu zobrazí hláška, že konfigurační soubor nebyl nalezen nebo je nedostupný. Program uživateli umožní použití výchozího nastavení, toto také vytvoří nový konfigurační soubor s výchozími hodnotami.



Obr. 12 Dialog nastavení

Konfigurační soubor je spravován třídou `settings` tato třída je implementována podle návrhového vzoru jedináček obsahuje veškeré funkce pro načtení, uložení nebo změnu konfiguračního souboru. Pro vytvoření bylo využito meziplatformní knihovny `simpleini`. Tato knihovna je velmi jednoduchá na použití a plně dostačuje potřebám aplikace. Při používání byl zjištěn nedostatek, že při načítání neexistující nebo chybějící sekce knihovna nevrací žádnou chybu, ale vrací hodnotu 0.

Což znamená, že se nic závažného nestane, pouze se v nastavení objeví všechny parametry nastavené na 0.

Pro změnu nastavení bylo využito třídy *QDialog*. Tato třída obsahuje jednoduché dialogy, ale pro naše potřeby byl vytvořen vlastní dialog. Okno dialogu je implementováno ve třídě *SettingsDialog*, která dědí *QDialog*. Od *QDialogu* dědíme hlavně proto, že vytváří modální okno. To znamená, že je to jediné okno programu, které je v ten moment použitelné.

## 6.2.6 Tabulky

Program obsahuje 2 tabulky. Tabulka v pravém okraji programu, která obsahuje id pacienta a do jaké třídy je zařazen podle vybraných metod a druhá tabulka zobrazuje

```
QVariant data(const QModelIndex &index, int role) const;
QVariant CPatientsModel::headerData(int section, Qt::Orientation orientation, int role) const;
int rowCount(const QModelIndex &parent) const;
int columnCount(const QModelIndex &parent) const;
```

*Ukázka kódu 1 Metody, které je nutné přepsat při vytváření vlastního modelu*

výsledky vypočtených metod pro všechny pacient. Tabulky jsou vytvořeny jako *QTableView*, která využívá model/view architektury viz 6.6. Proto vytváření tabulky v *Qt* je nutné nejprve vytvořit model a ten poté přiřadit k dané tabulce. V programu je pro každou tabulku vytvořen vlastní model. Aby se třída dala využít, jako model musí rozšířit odpovídající model, v tomto případě *QAbstractTableModel*. Při dědění této třídy je nutné přepsat několik funkcí viz ukázka kódu 1.

```
if (orientation == Qt::Vertical){
    switch (section)
    {
        case 0:
            return tr("Median difference (x)");
        case 1:
            return tr("Linear least squares (x)");
        case 2:
            return tr("C*e^(k*x) (x)");
        case 3:
            return tr("Standard deviation (y)");
        case 4:
            return tr("Difference standard deviation offset (y)");
        case 5:
            return tr("Interquartile range (y)");
        default:
            break;
    }
}
```

*Ukázka kódu 2 Vytvoření vertikální hlavičky tabulky s modelem CPatientScore*

Funkce `data(...)` slouží k zobrazení dat mimo hlavičky. Také je v této funkci možné určit, jaké bude mít pole barvu atd. Funkce `headerData(...)` umožňuje nastavit hlavičky jednotlivých řádků (viz ukázka kódu 2) a sloupců. Funkce `rowCount(...)` a `columnCount(...)` určují celkový počet řádek a sloupců.

Informace  
 Výsledky metod: Konvergence/Variabilita  
 Konvergence: L - Lepší, H - Horší  
 Variabilita: L - Lepší, H - Horší  
 Konvergence: Medián diferencí  
 Variabilita: Mozikvartilové rozpětí  
 Hraniční bod: Průměr  
 Offset: 15

| ID pacienta | Výsledky metod |
|-------------|----------------|
| 1           | H/L            |
| 2           | L/L            |
| 3           | L/L            |
| 4           | L/H            |
| 5           | L/L            |
| 6           | H/H            |
| 7           | H/L            |

Obr. 14 Ukázka tabulky s modelem `CPatientsModel`

První z tabulek má model spravovaný třídou `CPatientsModel` tato tabulka zobrazuje, do jaké třídy je každý pacient zařazen ve formátu konvergence/variabilita. Tato tabulka obsahuje 2 sloupce a počet řádků je závislý na počtu načtených pacientů ze souboru. Druhý sloupec poté obsahuje, do jaké skupiny byli pacienti zařazeni ve formátu konvergence/variabilita. Z této tabulky je poté možné vybrat jednotlivé řádky a zobrazit je ve formě grafu viz 7.2.4.

Druhá tabulka má model ve třídě `CPatientScore`. Tento model je o trochu složitější než u tabulky předchozí. Zobrazená data jsou výsledky jednotlivých metod pro všechny pacienty. Tabulka má vertikální i horizontální hlavičku. Horizontální hlavička obsahuje jména pacientů, kdežto ve vertikální hlavičce se nacházejí jména jednotlivých metod. Tento model má také podbarvená jednotlivá pole podle toho, zda je hodnota metody lepší, horší nebo rovna hraničnímu bodu. Pole obsahující lepší hodnoty jsou

obarveny zeleně, pole s horšími hodnotami mají barvu červenou a pole s hodnotami, které se rovnají hraničnímu bodu, jsou zobrazeny modře.

|  | 1           | 2           | 3           | ^  |
|--|-------------|-------------|-------------|----|
| Medián diferencí (x)                     | 1           | -1          | -1          |    |
| Lineární metoda nejmenších čtverců (x)   | 0.0751947   | -0.645384   | 0.202225    |    |
| $C \cdot e^{(k \cdot x)}$ (x)            | 0.000104261 | -0.00256985 | 0.000308388 |    |
| Standardní odchylka (y)                  | 5.69054     | 12.2084     | 5.28615     |    |
| Standardní odchylka diferencí offset (y) | 4.7622      | 3.06394     | 6.54225     | .. |

Legenda

Barva lepších hodnot: Zelená

Barva horších hodnot: Červená

Barva středové hodnoty: Modrá

Další informace: U názvu metody je v závorce napsáno na jaké ose je metoda zobrazena.

Obr. 15 Ukázka tabulky s modelem CPatientScore

## 6.2.7 Export grafů do SVG

Aplikace umí vyexportovat grafy do formátu SVG (Scalable Vector Graphics). SVG je formát souboru, který popisuje dvojrozměrnou vektorovou grafiku pomocí xml (Extensible Markup Language). SVG je vhodné pro jednoduchou grafiku např grafy, binární stromy atd.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="209.903mm" height="297.039mm"
  viewBox="10 8 576 819"
  xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
  version="1.2" baseProfile="tiny">
<title>Graf4</title>
<desc>Generated with Qt</desc>
<defs>
</defs>
<g fill="none" stroke="black" stroke-width="1" fill-rule="evenodd" stroke-
linecap="square" stroke-linejoin="bevel" >

<g fill="none" stroke="#808080" stroke-opacity="0.392157" stroke-width="1" stroke-
linecap="square" stroke-linejoin="bevel" transform="matrix(1,0,0,1,9.6,11.34)"
font-family="Arial" font-size="30" font-weight="400" font-style="normal"
>
<text fill="#808080" fill-opacity="0.392157" stroke="none" xml:space="preserve"
x="143" y="102" font-family="Arial" font-size="30" font-weight="400" font-
style="normal"
>L/H</text>
<text fill="#808080" fill-opacity="0.392157" stroke="none" xml:space="preserve"
x="431" y="102" font-family="Arial" font-size="30" font-weight="400" font-
style="normal"
>H/H</text>
```

Ukázka kódu 3 SVG soubor, popis jednotlivých částí pomocí xml

V *Qt* se pro export do SVG využívá *QSvgGenerator*. Tomuto generátoru se nastaví velikost stránky. Poté se nastaví *viewBox*, což je prostor, na který vykreslí požadované prvky. Kreslení na plátno probíhá pomocí *QPainter*, kterému se nastaví *QSvgGenerator*.

```
QSvgGenerator generator;
generator.setFileName("filename.svg");
generator.setSize(QSize(500, 500));
generator.setViewBox(QRect(0, 0, 200, 200));
generator.setTitle(tr("SVG title"));
QPainter painter;
painter.begin(&generator);
....
painter.end();
```

*Ukázka kódu 4 Vytvoření SVG souboru a nastavení*

V programu se do SVG exportují oba grafy na stránku o velikosti A4. V horní polovině je graf rozdělení pacientů do skupin a ve spodní se nachází graf hodnot *HbA1c* pacientů. K tomu, aby grafy byly pod sebou, bylo využito translace o polovinu velikosti stránky.

Při vykreslování grafů byly 2 možnosti, použití *scale* nebo přepočítat souřadnice grafu na požadovanou velikost. *Scale* je na použití jednodušší a rychlejší. Problém nastal při zvětšení okna a následném zmenšení pomocí *scale* pro export se části grafu staly nečitelnými. Z tohoto důvodu bylo použito přepočítání souřadnic grafu na požadovanou velikost. Čímž je zajištěna konstantní velikost jednotlivých komponent.

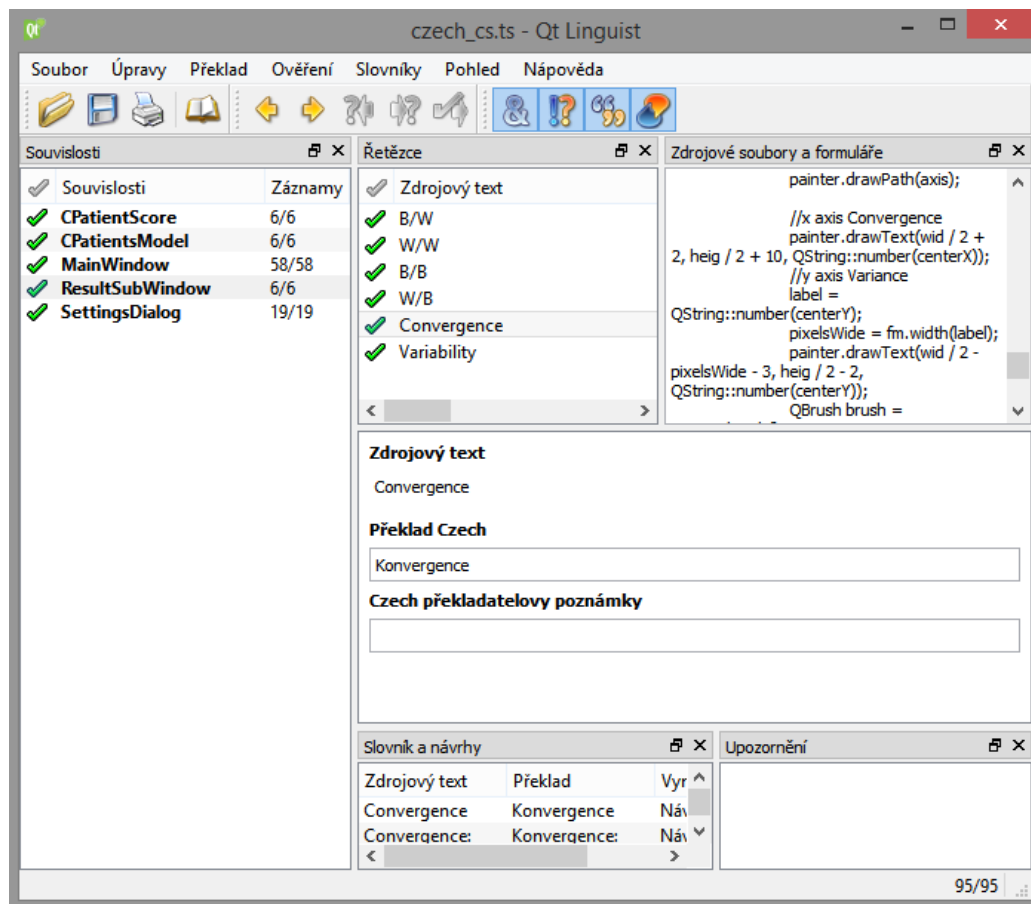


*Obr. 16 Použití scale v qt*

## 6.2.8 Překlad aplikace do češtiny

Pro překlad do češtiny bylo využito programu, který je součástí *Qt*. Program se jmenuje *Qt Linguist*, tento program je velmi jednoduchý na použití s *Qt* aplikací a proto byl zvolen. Tento program podporuje překlad do mnoha jazyků. *Qt Linguist* naskenuje veškeré třídy, které používají *Q\_Object* makro. Poté podle toho vytvoří soubor typu *ts* soubor, který obsahuje jednotlivé fráze použité v kódu. Text se bude vyskytovat v

překládovém souboru, pokud ho v kódu zapíšeme, jako `tr("text")`. Poté je možné tyto fráze přeložit v *Qt Linguist*.



Obr. 17 Okno programu *Qt Linguist*

Překládový soubor typu `ts` je ve formátu XML. To mu dodává přehlednost pro úpravu bez využití grafického prostředí. Tento soubor je automaticky vygenerován a aktualizován při použití `lupdate`. Následné vydání souboru se provede pomocí `lrelease`. Obě tyto funkce jsou zakomponovány v *Qt Creatoru* a pokud je nainstalován přídavek *Qt* do *Visual Studia* nalezneme tyto příkazy i tam.

```

QApplication a(argc, argv);

QTranslator translator;
translator.load("czech_cs");
a.installTranslator(&translator);

```

#### Ukázka kódu 5 Ukázka použití překládového souboru

Překládový soubor je poté nainstalován do aplikace pomocí příkazů viz ukázka kódu 5. Pokud by soubor nebyl nalezen, budou použity zdrojové texty.

Nevýhodou tohoto programu je, že změny v délce kódu se musí promítnout i do překladových souborů. To znamená, že po každé změně v délce kódu se musí volat *lupdate*, poté se musí otevřít *Qt Linguist* a musí se potvrdit všechny posunuté překlady, zda jsou správně, poté už překladový soubor opět funguje. Z tohoto důvodu by bylo možná časově méně náročné využití vlastní implementace překladového souboru.

```
<context>
  <name>CPatientsModel</name>
  <message>
    <location filename="cpatientsmodel.cpp" line="86"/>
    <source>Id patient</source>
    <translation></translation>
  </message>
  <message>
    <location filename="cpatientsmodel.cpp" line="90"/>
    <source>Method results</source>
    <translation>Výsledky metod</translation>
  </message>
</context>
```

Ukázka kódu 6 Překladový soubor

## 6.2.9 Výpočty metod

Pro výpočet metod slouží 2 třídy, každá se stará o jeden typ. Třída *ConvergenceMethod* se stará o výpočet metod konvergence a *VarianceMethod* se stará o výpočet variability. Postupy výpočtu viz kapitola 4.

Třída *ConvergenceMethod* využívá knihovny *Eigen*<sup>4</sup>. Tato matematická knihovna je využita k řešení soustav rovnic a metody nejmenších čtverců. Knihovna *Eigen* byla použita z důvodu spolehlivosti a její jednoduchosti na použití.

## 6.3 Výkonnost aplikace

Velmi důležitý parametr u všech aplikací je, jak rychle dokáže aplikace reagovat na uživatelský vstup a celkový plynulý běh aplikace. Toto bylo důležité i u této aplikace.

Testovací sestava: Notebook Lenovo Y580

- Procesor: Intel i5-3210M @2.5 GHz
- RAM: 8 GB
- Grafická karta: NVIDIA GeForce GTX 660M 2GB
- HDD: 1TB

<sup>4</sup> [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page)



- Operační systém: Windows 8.1

### 6.3.1 Načítání vstupu

Z počátku vývoje aplikace bylo načítání vstupu velmi pomalé. Bylo to způsobeno mou chybou při vytváření instance *QXlsx::document*. Toto způsobilo, že načítání souboru mohlo trvat i několik vteřin. Po odstranění této závady bylo načítání téměř okamžité. Rychlosti načítání na testovací sestavě viz tab. 2 jsou již dle mého názoru dostačující. Nejdelší dobu zabírá vytvoření modelů tabulek a jejich zobrazení v GUI.

Obsah testu:

- Načtení vstupního souboru
- Vypočet všech metod pro všechny pacienty
- Vypočet průměrů a mediánů pro všechny metody
- Vytvoření modelů tabulek
- Zobrazení tabulek

| Počet pacientů (30 hodnot každý) | Čas [s] |
|----------------------------------|---------|
| 20                               | 0.026   |
| 400                              | 0.132   |
| 1000                             | 0.324   |

*Tab. 2 Rychlost načítání vstupu a jeho zpracování*

### 6.3.2 Export do SVG

Toto je časově nejnáročnější funkce celé aplikace, zejména kvůli grafu hodnot pacientů, který byl optimalizován. Optimalizace spočívala ve snížení počtu přepočítávání křivek. Toto by bylo vhodné řešení, pokud by při exportu byl využit scale, ale vzhledem k tomu, že byla zvolena metoda přepočítávání bodů na zvolenou velikost, byla tato optimalizace spíše zpomalením, což bylo odhaleno později během vývoje. Tato funkce, by při dalším vývoji aplikace byla znovu optimalizována.

Obsah testu:

- Vytvoření *QSvgGenerator*

- Vypočítání translací
- Vykreslení grafu hodnot *HbA1c* pacientů
- Vykreslení grafu rozdělení do skupin pacientů

| Počet zobrazených hodnot | Čas [s] |
|--------------------------|---------|
| 20                       | 0.954   |
| 400                      | 1.412   |
| 1000                     | 2.273   |

Tab. 3 Rychlost exportu do SVG

Z výsledků (tab. 3) je vidět, že tato funkce je již docela časově náročná a zajisté by bylo dobré pokusit se čas trvání zkrátit.

### 6.3.3 Ukládání xlsx souboru

Při exportu xlsx souboru nejsou prováděny téměř žádné výpočty, proto tato funkce by podle předpokladu neměla trvat dlouhou dobu.

Obsah testu:

- Vytvoření xlsx souboru
- Vyplnění listu daty *HbA1c*
- Vyplnění druhého listu výsledky všech metod pro všechny pacienty
- Vykreslení grafu pacientů na poslední list

| Počet pacientů (30 hodnot každý) | Čas [s] |
|----------------------------------|---------|
| 20                               | 0.095   |
| 400                              | 1.449   |
| 1000                             | 3.057   |

Tab. 4 Doba uložení xlsx souboru

Tyto časy mě velmi překvapily, protože z mé strany zde nejsou žádné výpočty, pouze cykly pro naplnění jednotlivých listů odpovídajícími daty.

## 6.4 Vytvoření instalačního souboru

Pro aplikaci byl vytvořen instalační soubor. K vytvoření tohoto souboru byl využit program *Inno Setup*, tento skript po zkompilování v *Inno Setup* vytvoří

spustitelný instalační soubor. Při vytváření instalačního souboru *Qt* aplikace je nutné přidat několik knihoven a všechny soubory nutné ke spuštění aplikace. Poté stačí jen zkompilovat.

Program *Inno Setup* je jednoduchý na použití a obsahuje průvodce, který pomůže při generování skriptu. Úprava a napsání samotného skriptu, ale není obtížné i bez průvodce.

```
[Languages]
Name: "english"; MessagesFile: "compiler:Default.isl"
Name: "czech"; MessagesFile: "compiler:Languages\Czech.isl"

[Tasks]
Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked

[Files]
Source: "C:\Users\Cybra\Disk Google\untitled3\Release\untitled3.exe"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Users\Cybra\Disk Google\untitled3\config.ini"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Qt\Qt5.5.1\Tools\QtCreator\bin\Qt5Core.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Qt\Qt5.5.1\Tools\QtCreator\bin\Qt5Gui.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Qt\Qt5.5.1\Tools\QtCreator\bin\Qt5PrintSupport.dll"; DestDir: "{app}"; Flags: ignoreversion
```

*Ukázka kódu 7 Ukázka skriptu pro vytvoření instalačního souboru*

## 7 Analýza hodnot HbA1c

Po dokončení programu byla provedena datová analýza, ve které byly vyhodnoceny jednotlivé metody a jejich kombinace proti sobě. Dále byly připraveny data pro analýzu ANOVA, pro všechny kombinace metoda konvergence a variability a pro konvergenci a varianci samostatně.

Exportovaný soubor obsahuje několik listů. První list obsahuje seznam pacientů a kvadranty, v kterých se pacienti nacházejí pro veškeré kombinace metod se středem v mediánu, tento list je nazván "Median". Dále jsou listy s názvem "Compare 1" až "Compare 9", tyto listy obsahují porovnání všech kombinací s referenční metodou. Druhý sloupec v těchto listech obsahuje referenční hodnotu a zbylé sloupce kromě posledního obsahují porovnání s referenčním sloupcem. Hodnota 1 značí shodu kvadrantu, 0 neshodu. Poté následuje list "AnovaMedian", který obsahuje veškeré kombinace metod pro analýzu ANOVA. Dále následuje list "Convergence&Variability", tento list obsahuje seznam pacientů a zda jsou jejich hodnoty konvergence a variability větší 1 nebo menší 0 než medián. Listy s názvem "Convergence compare x" a "Variability compare x" obsahují porovnání výsledků jedné z metod metody s ostatními. Druhý sloupec těchto listů obsahuje referenční hodnotu a další kromě posledního obsahují porovnání s referenčním. Hodnota 1 značí shodu, 0 neshodu. List "ConvAnova" obsahuje všechny kombinace metod konvergence, připravené pro analýzu ANOVA. Poslední list "VarAnova" obsahuje kombinace metod variability pro analýzu ANOVA.

K datové analýze byl použit testovací soubor s 20 pacienty o 30 hodnotách pro každého pacienta. Tyto údaje byly vyhodnoceny všemi metodami pro výpočet z kapitoly 4. Pro konvergenci a variability bylo porovnáno, zda jsou data lepší nebo horší než medián a poté porovnáno v kolika případech se výsledek liší.

Pro kombinace metod konvergence a variability bylo použito stejných metod, ale, byl spočítán kvadrant, do kterého by tyto hodnoty patřili. Pokud je hodnota na ose, tak je přiřazena kvadrantu s horším ohodnocením. To znamená, že osy vymezující 1. kvadrant do něho také spadají. A naopak 3. kvadrant používá ostré nerovnosti.

| Referenční metoda              | Medián<br>diferencí | Jednoduchá lineární<br>regrese | $C^*e^{(K^*x)}$ | Shodných v<br>průměru |
|--------------------------------|---------------------|--------------------------------|-----------------|-----------------------|
| Medián diferencí               |                     | 0.65                           | 0.65            | 0.65                  |
| Jednoduchá lineární<br>regrese | 0.65                |                                | 1               | 0.825                 |
| $C^*e^{(K^*x)}$                | 0.65                | 1                              |                 | 0.825                 |

Tab. 6 Shoda referenční metody konvergence s ostatními

Při porovnání metod konvergence bylo zjištěno, že lineární metoda nejmenších čtverců a  $C^*e^{(K^*x)}$  vrací na testovaných datech vždy stejný výsledek. To znamená, že jednu z těchto metod je možné vypustit, doporučil bych vynechat  $C^*e^{(K^*x)}$  jejíž výsledky se pohybují v řádech  $10^{-3}$  a menší, z tohoto důvodu jsou dle mého názoru hodnoty špatně čitelné a tuto metodu bych odstranil. Na testovaných datech bylo zjištěno, že při porovnání metod dojde v průměru k 0.76666, shodě s referenční metodou.

| Referenční metoda<br>variability           | Standardní<br>odchylka | Standardní odchylka<br>diferencí s ofsetem | Mezikvartilové<br>rozpětí | Shodných |
|--|------------------------|--|---------------------------|----------|
| Standardní odchylka                        |                        | 0.6  | 0.8                       | 0.7      |
| Standardní odchylka<br>diferencí s ofsetem | 0.6                    |  | 0.7                       | 0.65     |
| Mezikvartilové<br>rozpětí                  | 0.8                    | 0.7  |                           | 0.75     |

Tab. 5 Shoda referenční metody variability s ostatními

Pro metody variability nebyly nalezeny výrazné rozdíly. Shoda na testovaných datech byla v průměru 0.7. Zajímavé je že standardní odchylka a Standardní odchylka diferencí s ofsetem je pro testovací data více rozdílné než mezikvartilové rozpětí. Ofset u standardní odchylky diferencí je použit k odstranění extrémních hodnot z počátku léčby.

Pro kombinace metod byl proveden stejný test. Výsledky testu shody s referenční metodou je v průměru 0.505556. Jisté kombinace metod obsahovaly velmi rozdílné výsledky. Byl zde velký rozdíl mezi shodou, kde metoda konvergence byla medián diferencí a zbylými dvěma z důvodů stejných výsledků metody nejmenších čtverců a  $C^*e^{(K^*x)}$ . Celkově je shoda kombinací metod celkem nízká. Pro některé

kombinace metod se shoda pohybovala, až kolem 0.2 což je velmi podstatný rozdíl v hodnocení.

| Referenční metoda   | Shoda s ostatními metodami |
|---|----------------------------|
| Medián diferencí + Standardní odchylka                                | 0.44375                    |
| Medián diferencí + Standardní odchylka diferencí s ofsetem            | 0.4                        |
| Medián diferencí + Mezikvartilové rozpětí                             | 0.41875                    |
| Jednoduchá lineární regrese + Standardní odchylka                     | 0.5625                     |
| Jednoduchá lineární regrese + Standardní odchylka diferencí s ofsetem | 0.53125                    |
| Jednoduchá lineární regrese + Mezikvartilové rozpětí                  | 0.55                       |
| $C * e^{(K * x)}$ + Standardní odchylka                               | 0.5625                     |
| $C * e^{(K * x)}$ + Standardní odchylka diferencí s ofsetem           | 0.53125                    |
| $C * e^{(K * x)}$ + Mezikvartilové rozpětí                            | 0.55                       |

*Tab. 7 Shoda kombinací metod*

## 8 Závěr

Při vypracovávání práce jsem se seznámil s problematikou lidí, kteří mají diabetes. Seznámil jsem se s metodami, kterými se měří glykovaný hemoglobin a v jakých hodnotách se uvádí. Dále jsem zjistil nebezpečí, kterým by se lidé trpící diabetem měli vyhnout a jaké hodnoty glykovaného hemoglobinu jsou v mezích, a kdy už hodnoty vypovídají o diabetu. Dále jsem se seznámil s prací v jednom z nejrozšířenějších multiplatformních frameworků pro vytváření grafických uživatelských rozhraní *Qt*.

Na základě těchto poznatků jsem navrhl a implementoval uživatelské rozhraní, které je snadno ovladatelné, toto jednoduché uživatelské rozhraní by mělo usnadnit uživatelům používání aplikace.

Dalším úkolem bylo vytvořit samotnou funkcionalitu programu. Byly implementovány všechny v práci uvedené výpočetní metody a veškeré popsané funkce. Výsledný program dokáže přehledně zobrazit informace jak v tabulkách, tak pomocí grafu.

Nepodařilo se mi nalézt aplikaci s podobnou funkcionalitou, proto nemohou porovnat můj softwarový produkt s ostatními.

Aplikace lze dále rozšířit o přepínání mezi různými jazyky. Přidání dalších kvalifikačních metod. Také by měla při dalším vývoji proběhnout další optimalizace některých součástí.

## Seznam zkratek

- HbA1c - Glykovaný hemoglobin
- NGSP - National Glycohemoglobin Standardization Program
- IFCC - International Federation of Clinical Chemistry
- DCCT - The Diabetes Control and Complications Trial
- GUI - Graphical User Interface
- OSS - Open-Source Software
- QML - Qt Modeling Language
- MVC - Model-View-Controller
- IDE - Integrated Development Environment
- XML - Extensible Markup Language
- SVG - Scalable Vector Graphics



## Zdroje

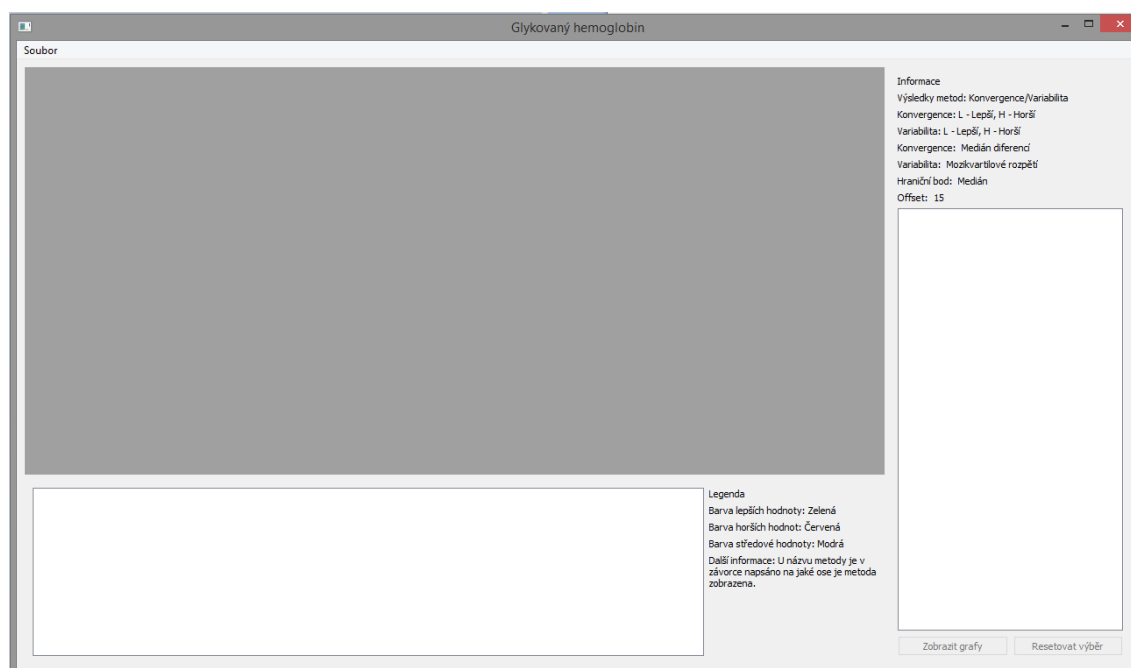
1. AMERICAN DIABETES ASSOCIATION, et al. 2. Classification and diagnosis of diabetes. *Diabetes Care*, 2015, 38.Supplement 1: S8-S16
2. AMERICAN DIABETES ASSOCIATION, et al. Diagnosis and classification of diabetes mellitus. *Diabetes care*, 2013, 36.Supplement 1: S67-S74.
3. Ústav zdravotnických informací a statistiky, *Péče o nemocné s cukrovkou*, Praha, 2012, 978-80-7472-082-6
4. KERNER, W.; BRÜCKEL, J. Definition, classification and diagnosis of diabetes mellitus. *Exp Clin Endocrinol Diabetes*, 2014, 122.7: 384-6.
5. REIF, Jiří. *Metody matematické statistiky*. Západočeská univerzita, 2004
6. HINTON, Perry R. *Statistics explained*. Routledge, 2014
7. A. C. Guyton and J. E. Hall, *Medical Textbook of Physiology*, Philadelphia: Elsevier Inc., 2006.
8. D. Longo, A. Fauci, D. Kasper, S. Hauser, J. Jameson and J. Loscalzo, *Harrison's principles of internal medicine*, New York: Mc Graw-Hill, 2011.
9. Blanchette, Jasmin and Mark Summerfield. *C++ GUI programming with Qt 4. 2nd ed.*, Upper Saddle River, NJ: Prentice Hall in association with Trolltech Press, 2008.

# Příloha A: Uživatelský manuál

## Instalace

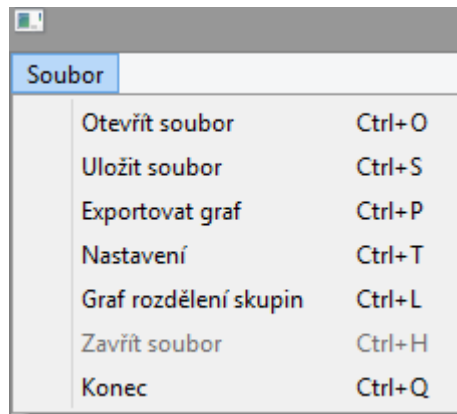
Pro snadnou instalaci na operačním systému Windows byl vytvořen jednoduchý instalátor s názvem "setup.exe". Po dokončení instalace můžeme program spustit buď zástupcem na ploše, nebo spustitelným souborem ve složce, do které byl program nainstalován v základu "C:\Program Files (x86)\GlykHem\untitled3.exe".

Pokud nemáme instalační soubor, ale máme dostupný samotný projekt, je nutné stáhnout veškeré potřebné komponenty, tedy Qt verze 5.5 a výš, simpleini, QtXlsx a Eigen. Po instalaci všech těchto součástí můžeme program přeložit a spustit.



Obr. A 1 Okno aplikace po spuštění

## Hlavní menu



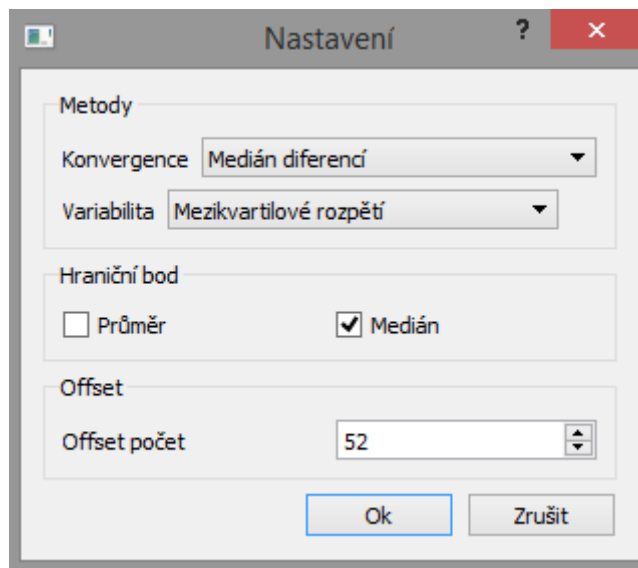
*Obr. A 2 Hlavní menu*

Hlavní menu obsahuje většinu funkcionalitu programu. Toto menu lze otevřít kliknutím na nápis soubor.

- Otevřít soubor - načte excelovský soubor (xlsx), který zvolíme ve výběrovém dialogu. Zobrazí se tabulky s daty
- Uložit soubor - Uloží načtená data do excelovského souboru (xlsx), původní data, vypočtené metody a graf
- Exportovat graf - Exportuje oba grafy do SVG souboru, název a umístění je zvoleno v dialogovém okně, které se objeví po použití tohoto tlačítka
- Nastavení - Otevře dialog nastavení
- Graf rozdělení skupin - zobrazí graf pacientů rozdělených do skupin podle zvolených metod v nastavení
- Zavřít soubor - Smaže veškeré otevřené grafy, smaže obě tabulky a nastaví GUI do výchozího vzhledu
- Konec - Ukončí program

## Nastavení

Dialog nastavení slouží k úpravě nastavení programu



Obr. A 3 Dialog nastavení

Metody:

- Konvergence: nastaví, která metoda konvergence se má použít
- Variabilita: nastaví, která metoda variability se má použít

Hraniční bod:

- Určuje, zda se jako hraniční bod použije průměr hodnot nebo medián

Offset:

- Nastavení kolik hodnot má vynechat metoda standardní odchylky s ofsetem
- Hodnoty od 0

Při stisknutí obou tlačítek okno zmizí, ok změny potvrdí a provede, cancel zruší zadané změny.

## Vykreslení grafů

|    |     |
|----|-----|
| 11 | H/L |
| 12 | H/L |
| 13 | H/L |
| 14 | H/L |
| 15 | H/H |
| 16 | L/L |
| 17 | H/H |
| 18 | H/L |

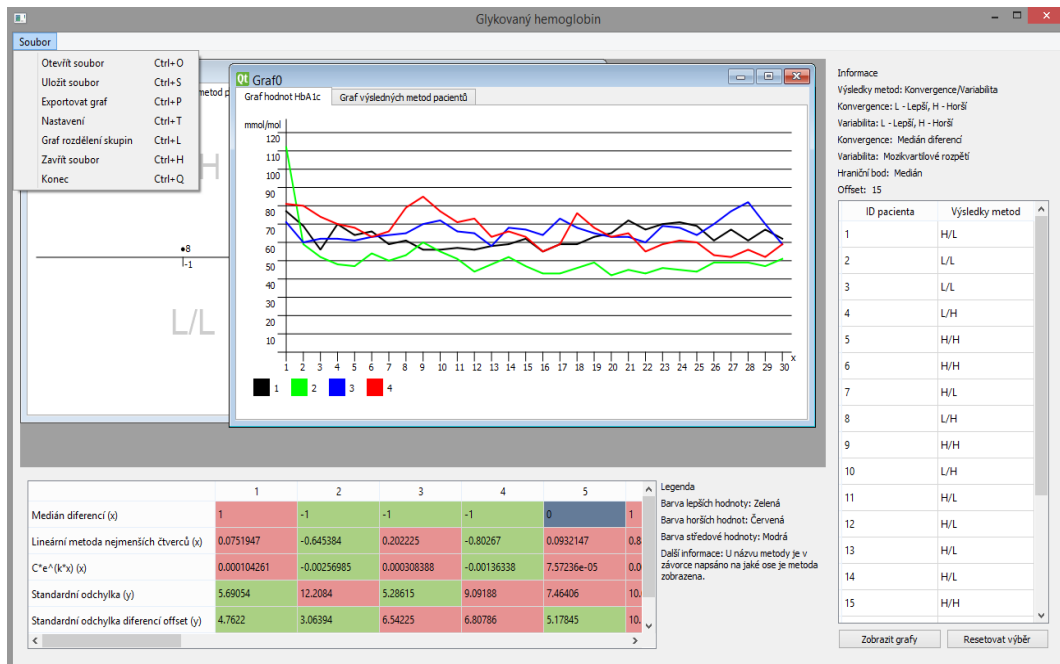
Zobrazit grafy      Resetovat výběr

*Obr. A 4 Tabulka a tlačítka pro vykreslení grafu*

Pro vykreslení grafů zvolíme požadované pacienty z tabulky při pravém okraji aplikace. A poté zmáčkneme zobrazit grafy. Po zmáčknutí tohoto tlačítka budeme vyzváni k zadání názvu grafu. V základu je tento název "Graf + číslo", toto okno má 2 tlačítka ok a cancel, ok potvrdí název a zobrazí okno s grafy. Cancel naopak zruší vytváření grafů.

Tlačítko resetovat výběr odznačí zvolené řádky v tabulce.

## Celkový vzhled aplikace



Obr. A 5 Celkový vzhled aplikace s otevřeným souborem a oběma typy grafů

