



CERTIFIKOVANÁ METODIKA

Úprava soustav rovnic pro předpověď genomické plemenné hodnoty (GEPH) jednokrokovou metodou

Autoři

**prof. Ing. J. Příbyl, DrSc., Mgr. Ing. J. Bolečková, Mgr. J. Haman,
Ing. T. Kott, PhD., Ing. J. Příbylová, CSc., RNDr. M. Šimečková, Dr. nat. techn.,
Ing. L. Vostrý, PhD., Ing. L. Zavadilová, CSc.**

Oponenti

Ing. Zdenka Majzlíková
Česká plemenářská inspekce

doc. Ing. Juraj Candrák, PhD.
Slovenská poľnohospodárska univerzita v Nitre

Zpracováno v rámci řešení výzkumného projektu NAZV č. QI111A167

ISBN 978-80-7403-079-6

Česká plemenářská inspekce

v y d á v á

OSVĚDČENÍ

1877/2011-ČPI

o uznání uplatněné certifikované metodiky
v souladu s podmínkami „Metodiky hodnocení výsledků výzkumu a vývoje“

**Úprava soustav rovnic pro předpověď genomické plemenné hodnoty (GEPH)
jedenkrokovou metodou**

J. Příbyl, J. Bolečková, J. Haman, T. Kott, J. Příbylová, M. Šimečková, L. Vostrý,
L. Zavadilová

Výzkumný ústav živočišné výroby, v.v.i., Přátelství 815, 10400 Praha Uhřetěves

Vypracované v rámci výzkumného projektu/záměru č. NAZV QI111A167.

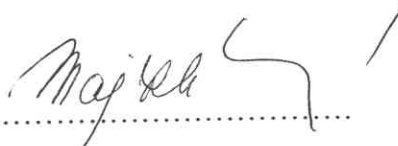
V Praze dne 9.6.2011

(Razítko odborného orgánu státní správy):

Česká plemenářská inspekce
P. O. BOX 774
Štěpánská 63
111 21 Praha 1
1

(Jméno a funkce zástupce odborného útvaru státní správy): Ing. Zdenka Majzlíková
ředitelka

(Podpis zástupce odborného útvaru státní správy):


.....

OBSAH

I. Cíl metodiky	4
II. Vlastní popis metodiky	4
II.1. Úvod	4
II.2. Pracovní postup	5
II.3. Ověření pracovních postupů a algoritmy	8
III. Srovnání Novosti postupů	9
IV. Popis Uplatnění certifikované metodiky	9
V. Ekonomické aspekty	9
VI. Seznam použité související literatury	10
VII. Seznam publikací, které předcházely metodice	12
Příloha 1	13
Program v SASu na stanovení a porovnání GEPH maticovým výpočtem	13
Příloha 2	25
Úsek programu v SASu pro sestavení G	25
Příloha 3	27
Úsek programu v SASu pro sestavení A_{22}	27
Příloha 4	28
Algoritmy načítání velkých souborů a průběžné tvorby součtů	28
Příloha 5	30
Popis věty výstupu z laboratoře (část)	30

I. CÍL METODIKY

Cílem této metodiky je postup úpravy soustav rovnic animal modelu pro předpověď plemenné hodnoty (PH) na jednokrokovou metodu předpovědi (single step prediction (SSP)) genomické plemenné hodnoty (GEPH). Spočívá v úpravě inverze rodokmenové matice příbuznosti mezi zvířaty pomocí genomické matice příbuznosti. Genomická matice příbuznosti je stanovena na základě SNP (single nucleotide polymorphism) markerů, které hustě pokrývají celý genom. GEPH slouží především k výběru mladých zvířat bez známé vlastní užitkovosti.

II. VLASTNÍ POPIS METODIKY

II.1. Úvod

Začlenění molekulárně-genetických (MG) informací do hodnocení plemenných zvířat prodělává poměrně prudký vývoj.

Toto začlenění bylo nejdříve prostřednictvím vztahu několika genetických markerů ke QTL (quantitative trait locus) chovatelsky významných vlastností (Příbyl 1995). Údaje o MG doplňovaly PH stanovené pomocí běžných postupů metody BLUP- animal model. Doplnění bylo buď cestou začlenění genetických markerů přímo do modelových rovnic odhadu PH, nebo prostřednictvím oddělených výpočtů. Dílčí PH na základě markerů a polygenní PH byly pomocí vah sloučeny do selekčního indexu.

S vývojem laboratorních postupů je možné dnes stanovit velké množství genetických markerů. Tím jsou podmíněny i postupy práce při předpovědi PH. Postupně se přešlo na metody, které odhadují velký počet regresních závislostí mnoha QTL na mnoha markerech SNP (single nucleotide polymorphism). Těchto mnoho regresních koeficientů se následně sloučí pomocí selekčního indexu do jednoho ukazatele, kterým se předpoví přímá genetická hodnota (PGH) (direct genetic value (DGV)) a ve spojení se „zbytkovou“ polygenní PH, stanovenou běžným způsobem pomocí animal modelu, opět pomocí selekčního indexu, se stanoví genomická plemenná hodnota (GEPH) (genomic breeding value (GEBV)) hodnoceného zvířete (Meuwissen et al. 2001). Pracujeme se šlechtěnými populacemi, které se trvale vyvíjí, rovněž se mění odezva genotypu na trvale se měnící podmínky chovu. Proto je nutné regresní koeficienty trvale upřesňovat, u skotu nejdéle ve dvouletých obdobích, nejlépe každoročně (Schaeffer 2006).

Práce s velkým počtem těchto genetických regresních koeficientů vede, jak bylo výše uvedeno, k vytvoření jedné souhrnné hodnoty pro každého jedince, na základě které se rozhodne o zařazení do plemenitby. Dílčí regresní koeficienty a jednotlivé genetické markery, proto přestávají být pro šlechtitele zajímavé a důležitý je především souhrnný ukazatel GEPH.

Genetické markery SNP mohou posloužit jak ke stanovení genetických regresních koeficientů, tak jsou základem ke stanovení realizované genomické příbuznosti G (Guo 1996). Genomickou a rodokmenovou příbuznost lze kombinovat (Bömcke a Gengler 2009, Bömcke et al. 2009).

VanRaden (2008) prokázal, že stejné GEPH ohodnocení jedinců se docílí přes výše zmíněné regresní koeficienty, nebo prostřednictvím G . Stanovení GEPH prostřednictvím G je výpočetně jednodušší.

Dosavadní pracovní postupy vycházely z dostatečně velkých referenčních souborů plemeníků se známými a vysoce spolehlivými plemennými hodnotami, u kterých se stanovily výše zmíněné regresní koeficienty. Jednalo se, pokud možno o věrohodné vzorky vybrané z celé hodnocené populace, aby byly představitelky jak průměru, tak aby pokryly celé rozpětí proměnlivosti

základního souboru. Podkladem pro stanovení regresních koeficientů byly, v návaznosti na známé PH těchto plemenů, zpětně stanovené opravené odchylky užítkovosti dcer DYD (daughter yield deviation) a jejich spolehlivosti vyjádřené v EDC (effective daughter contribution) (Szyda et al. 2008, VanRaden a Wiggans 1991). Hodnoty mladých ogenotypovaných zvířat byly následně stanoveny dosazením těchto regresních koeficientů ke stejným SNP u mladých jedinců. Nebo byla hodnota mladých zvířat stanovena prostřednictvím **G** na základě příbuzenského vztahu mezi ogenotypovanými starými a mladými zvířaty (Hayes et al. 2009, VanRaden et al. 2009).

Jednalo se o navazující víceřadové výpočty, každý krok vyžadoval splnění řady předpokladů a pokud možno bezchybné vstupní parametry. Víceřadové řešení vnáší do hodnocení další chyby (Verbyla et al. 2010). Splnit jednotlivé vstupní předpoklady při víceřadovém postupu je obtížné a dochází k předselekcí údajů, neboť bývají genotypováni jen někteří jedinci, většinou s vyššími PH. Porovnání ogenotypovaných a neogenotypovaných jedinců v jednom celostátním žebříčku vyžaduje následné opravy jak průměrů, tak proměnlivosti jedinců s GEPH. Navíc hodnocení ogenotypovaných jedinců a jejich neogenotypovaných vrstevníků a ostatních příbuzných v základní populaci, se navzájem ovlivňují (Ducrocq 2011).

Uvedené obtíže překonává jednokrokové (SSP) hodnocení GEPH (Misztal et al. 2009, Christensen a Lund 2010), které pracuje přímo s celými základními soubory kontroly užítkovosti. Tento přístup vychází z obvyklých postupů práce s lineárními smíšenými modely (Příbyl a Příbylová 2002) a bezprostředně navazuje na veškeré zpracované a programově připravené celostátní způsoby genetického hodnocení zvířat (Plemdat 2011).

Metodické zásady stanovení GEPH a jejich vývoj jsou shrnuty v Hayes et al. (2009), Poschadel a Mayer (2011), Příbyl et al. (2010a,b,c), VanRaden et al. (2009). Nové poznatky v použití jednokrokového stanovení GEPH uvádějí Aguilar et al. (2010a), Misztal et al. (2010), Tsuruta et al. (2010), Forni et al. (2011) a Chen et al. (2011). Použitím těchto postupů stoupají nároky na výpočty. Jsou proto vyvíjeny nové výpočetní postupy a algoritmy programování (Aguilar a Misztal 2008, Misztal et al. 2009, Aguilar et al. 2010b).

SSP umožňuje upřesnit genetické hodnocení, jak ogenotypovaných, tak neogenotypovaných jedinců v širokých souvislostech hodnocené populace a ukazuje se výhledovou metodou pro budoucnost. Tento způsob hodnocení se postupně připravuje i v zahraničí (Misztal 2011).

II.2. Pracovní postup

Obvykle je hodnocení zvířat na základě PH stanovované pomocí animal modelu podle modelové lineární rovnice:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{b} + \mathbf{Z} \cdot \mathbf{u} + \mathbf{e} \quad (1)$$

kde: **Y** je vektor naměřených užítkovostí (závisle proměnná),
X, Z jsou matice vysvětlujících proměnných přiřazující naměřené užítkovosti pevným a náhodným efektům,
b, u jsou odhadované neznámé pevné a náhodné efekty,
e je vektor nepředvídatelných náhodných chyb.

S náhodnými efekty **u** souvisí kovarianční matice **A**, vyjadřující proměnlivosti těchto náhodných efektů.

V návaznosti na modelovou rovnici je sestavena soustava normálních rovnic. V jednodušších případech, při pouze jedné hodnocené vlastnosti a pouze jednom pevném a náhodném efektu lze soustavu zapsat:

$$\begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z} + k\mathbf{A}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{b} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{X}'\mathbf{Y} \\ \mathbf{Z}'\mathbf{Y} \end{pmatrix} \quad (2)$$

kde: k je poměr rozptylů reziduálního a daného náhodného efektu \mathbf{u} ,

\mathbf{A} je-li náhodným efektem genetický efekt jedince (plemenná hodnota), je \mathbf{A} aditivní maticí příbuznosti mezi jedinci, pokrývající i generace předků.

\mathbf{A} je stanovována na základě rodokmenových údajů. Obvykle se používají jedinci s užitkovostí a 4 generace jejich předků. K sestavení \mathbf{A} lze využít několik postupů (Jakubec et al. 1999, Mrode 2005). Prvky matice příbuznosti \mathbf{A} mají na diagonále hodnotu 1, pokud není jedinec výsledkem připařování příbuzných jedinců. Nebo hodnoty vyšší než 1, pokud se jedná o jedince z příbuzenské plemenitby. Vzhledem k poměrně malému počtu zohledněných generací předků (4 generace), nebývá obvykle příbuzenská plemenitba zjištěna a tudíž ani uvažována a všechny hodnoty na diagonále jsou 1. Mimo diagonálu jsou aditivní příbuznosti mezi jedinci. K sestavení \mathbf{A}^{-1} se používá algoritmů, které tvoří přímo \mathbf{A}^{-1} postupným přičítáním koeficientů na jednotlivá políčka matice, aniž je třeba sestavovat \mathbf{A} , a tu následně invertovat (Henderson 1975, Misztal a Gianola 1987).

Řešením uvedené soustavy rovnic se získá odhad zohledněných efektů \mathbf{b} , \mathbf{u} , v tomto případě

$$\mathbf{PH} = \mathbf{u} \quad (3)$$

kde: \mathbf{PH} je vektor plemenných hodnot všech jedinců zahrnutých do \mathbf{A} .

Vzhledem k náhodné mendelistické kombinaci úseků DNA při zrání pohlavních buněk a náhodnému mendelistickému výběru rodičovských pohlavních buněk při oplodnění (mnoho spermií, z nichž jedna oplodní vajíčko) jsou hodnoty v \mathbf{A} náhodnými veličinami, které mají proměnlivost (Guo 1996, Příbyl et al. 2010b). To znamená, že v \mathbf{A} jsou pouze očekávané hodnoty, které se mohou lišit od skutečné genetické příbuznosti mezi jedinci. Skutečná genetická příbuznost je uváděna v \mathbf{G} .

$$\mathbf{E}(\mathbf{G}) = \mathbf{A} \quad (4)$$

kde: \mathbf{G} je genomická matice příbuznosti mezi sledovanými jedinci.

Skutečnou, (realizovanou, genomickou) příbuznost lze stanovit pomocí molekulárně-genetických metod z podobnosti jednotlivých částí DNA příbuzných jedinců. Pokud se jedinci shodují ve větším podílu svých genomů, jsou si více příbuzní. Prvky \mathbf{G} mohou nabývat hodnot kladných, nuly, nebo záporných, s celkovým rozpětím od -1 do +1. To znamená, že jedinci si jsou geneticky do různého stupně podobní (hodnota prvku kladná), nebo na sobě nezávislí (hodnota prvku = 0), či k sobě navzájem opační (závislí záporně, hodnota prvku záporná), když nesou v jednotlivých lokusech DNA právě opačné alely. Na rozdíl od \mathbf{G} , jsou v rodokmenové matici \mathbf{A} pouze hodnoty nezáporné. V porovnání s genomickou příbuzností \mathbf{G} , je proto \mathbf{A} deformovaná.

Matice \mathbf{G} pokrývá pouze genotypované jedince, matice \mathbf{A} zahrnuje všechny jedince v hodnocené populaci. Pomocí \mathbf{G} lze opravit \mathbf{A} a tím upřesnit řešení soustavy (2). V tom případě je na místo \mathbf{A}^{-1} dosazeno \mathbf{H}^{-1} :

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{pmatrix} \quad (5)$$

kde: **H** je opravená matice příbuznosti,

$\begin{bmatrix} \dots \end{bmatrix}$ je rozměru jako **A** s rozdělením na 4 části.,

F je oprava na genomickou příbuznost, týká se pouze genotypovaných jedinců, je rozměru jako **G**. Je stanovována dvojím možným způsobem:

Misztal et al (2009,.....):

$$\mathbf{F} = \lambda \cdot (\mathbf{G}^{-1} - \mathbf{A}^{-1}_{22}) \quad (6)$$

kde: λ je váha (0 až 1), která se dává genomické příbuznosti. Předpokládá se, že **G** plně nepokryje celou genetickou proměnlivost hodnocené vlastnosti a stále zbývá část proměnlivosti na polygenní efekt. Proto jsou za λ dosazovány v různých případech hodnoty 0,7 až 1.

\mathbf{A}_{22} je aditivní matice příbuznosti na základě rodokmenu pouze pro genotypované jedince.

Matice \mathbf{A}_{22} je vyříznuta z matice \mathbf{A}^* . Matice \mathbf{A}^* je matice rodokmenové příbuznosti sestavená pouze na základě genotypovaných jedinců a požadovaného počtu generací jejich předků.

Pro výpočet **F** je třeba invertovat **G** a invertovat \mathbf{A}_{22} .

Christensen a Lund (2010) (program DMU, Madsen et al.2010....):

$$\mathbf{F} = ((1-w) \cdot \mathbf{G} + w \cdot \mathbf{A}_{22})^{-1} - \mathbf{A}^{-1}_{22} \quad (7)$$

kde: w je váha (0 až 1), která se dává původní rodokmenové příbuznosti. Určitá malá váha (například $w=0,01$) se dává v každém případě, neboť **G** samostatně nemusí jít invertovat. Pro přípravu obecných programů je výhodnější dát matici **G** samostatně váhu w_1 a matici \mathbf{A}_{22} váhu w_2 , jejichž součet se nemusí vždy rovnat 1.

Matice **G** a \mathbf{A}_{22} jsou ve velikostech, které lze invertovat. Při velkém počtu genotypovaných jedinců lze matice **G** a \mathbf{A}_{22} rozložit na části a celkovou inverzi udělat násobením a inverzemi dílčích částí, například podle následujícího postupu:

M je matice rozložená na části

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \quad (8)$$

Inverze potom je :

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{M}^{-1}_{11} + \mathbf{L}\mathbf{P}^{-1}\mathbf{L}' & -\mathbf{L}\mathbf{P}^{-1} \\ -\mathbf{P}^{-1}\mathbf{L}' & \mathbf{P}^{-1} \end{bmatrix} \quad (9)$$

kde: $\mathbf{L} = \mathbf{M}^{-1}_{11}\mathbf{M}_{12}$

$$\mathbf{P}^{-1} = (\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{L})^{-1}$$

V současnosti je pro skot u hodnoceného jedince stanovováno pomocí čipů o různé hustotě velké množství genetických markerů SNP. Nejčastěji se používá čip Illumina 50K s 54609 lokusy SNP. Popis výstupu z laboratoře je uveden v příloze 5. Takto husté pokrytí genomu se využívá k vytvoření **G**.

Na základě alel uvedených ve sloupcích 9 a 10 s označením A, B v příloze 5, je každému lokusu SNP přidělena číselná hodnota. Ta vyjadřuje kolik druhých alel je v daném lokusu.

Jsou-li přítomny obě první alely pak hodnota	0,	(homozygot),
jedna první a jedna druhá alela	1,	(heterozygot),
obě druhé alely	2,	(homozygot).

Z uvedených hodnot se sestaví matice $\mathbf{T}_{(n,m)}$, která má tolik řádků, kolik je genotypovaných zvířat (n) a tolik sloupců, kolik je sledovaných lokusů ($m = 54609$).

V jednotlivých lokusech (j) se spočtou přes všechna genotypovaná zvířata průměrné četnosti druhé alely (q_j), to znamená četnosti podle jednotlivých sloupců (j) matice \mathbf{T} . Na základě průměrných četností se stanoví matice průměrů \mathbf{Q} o rozměru stejném jako \mathbf{T} . Prvky v daném sloupci (j) matice \mathbf{Q} jsou stejné a podle jednotlivých sloupců mají hodnotu $2 \cdot q_j$. Tyto průměry se odečtou od prvků matice \mathbf{T} , takže se následně pracuje s odchylkami výskytu alel od průměrů.

Genomická matice příbuznosti potom je:

$$\mathbf{G} = \frac{(\mathbf{T} - \mathbf{Q})(\mathbf{T} - \mathbf{Q})^c}{\text{stopa}((\mathbf{T} - \mathbf{Q})(\mathbf{T} - \mathbf{Q})^c) / n} \quad (10)$$

kde: n je počet genotypovaných jedinců

Některé SNP v matici \mathbf{T} jsou vedeny jako chybějící údaje. Proto jednotlivé prvky výsledné matice \mathbf{G} mohly vzniknout na základě odlišných počtů násobení. Aby byly všechny prvky navzájem porovnatelné, je třeba udělat u každého prvku přepočítání na jednotný počet SNP (příloha 2).

Čtvercová matice \mathbf{G} má rozměr podle počtu genotypovaných jedinců. Je proto menší, než je počet SNP. Je normalizována na základě stopy čitatele (10), aby průměr diagonálních prvků \mathbf{G} byl 1, podobně jako u rodokmenové matice příbuznosti \mathbf{A} , ve které prvky diagonály jsou nezáporné, ale většinou rovné 1.

Vzhledem k tomu, že se ve (2) používá matice \mathbf{H}^{-1} , ale vazba mezi zvířaty je na základě přímo \mathbf{H} , projeví se oprava příbuznosti vkládaná pomocí \mathbf{G}^{-1} v širších souvislostech i u negenotypovaných jedinců a může ovlivnit jejich PH, jakož i PH vrstevníků. Tím odpadá krok výpočtu související se vkládáním SNP u negenotypovaných příbuzných jedinců (imputation), který se dělá při jiných způsobech výpočtu.

II.3. Ověření pracovních postupů a algoritmy

Pro ověření způsobu hodnocení byl sestaven vlastní program v maticovém jazyce IML/SAS (příloha 1). Z jednotlivých částí programu jsou zřejmé algoritmy. Program vytváří jednotlivé matice podle (2), sestavuje \mathbf{G} a \mathbf{A}_{22} , sestavuje soustavy rovnic, soustavy řeší, stanovuje PH a porovnává jednotlivé způsoby výpočtu.

Výpočty tímto vlastním programem na umělém souboru byly porovnány s běžně světově používanými programy ze skupiny BLUPF90 (Misztal et al 2002) a programem DMU (Madsen a Jensen 2008, Madsen 2010, Madsen et al. 2010). Výpočty se shodovaly.

Dále byly výpočty ověřeny na dvou souborech o přibližně 800.000 prvních laktací krav s 800 plemeníky o genotypovanými 50K čipy a rodokmenem zahrnujícím celkem 1.500.000 jedinců. Přípravy souborů pro výpočty a zpracování výsledků byly udělány vlastními programy v programu SAS, které jsou na vyžádání u autorů. Pro sestavení \mathbf{G} a \mathbf{A}_{22} byly použity algoritmy podle příloh 2 a 3.

Na těchto souborech se výsledky výpočtů běžné PH (animal model) programy BLUPF90 a DMU shodovaly. Výsledky PH a GEPH byly porovnány. Časy výpočtu PH oběma programy byly podobné. Čas výpočtu GEPH oproti PH, obojí programem DMU, byl mnohonásobně delší.

V příloze 2 je část SAS programu pro sestavení **G**, kde část práce je udělána v databázích a část v maticích. Rychlost sestavení závisí na použitém algoritmu, v programu jsou uvedeny dva algoritmy.

V příloze 3 je část SAS programu v maticové algebře pro sestavení **A₂₂** podle Quaase (1976).

V příloze 4 jsou algoritmy v SASu, které lze využít při čtení velkých souborů a tvorbě průběžných součtů.

III. SROVNÁNÍ NOVOSTI POSTUPŮ

Jedná se o nové pracovní postupy, které byly používány výzkumně, ale nejsou zatím v praktickém šlechtění využity. Postupy vedou k upřesnění předpovědi GEPH především u mladých zvířat bez známé vlastní užitkovosti před jejich zařazením do plemnitby. Tímto způsobem, na rozdíl od jiných postupů, které jdou cestou samostatného hodnocení přes vybrané vzorky referenční populace, nedochází ke ztrátám informací, je lepší porovnání s ostatní populací a zpětně jsou upřesněny předpovědi plemenných hodnot i pro negenotypované jedince v základní populaci. Předkládané postupy jsou jednodušší z hlediska úpravy počítačových programů a uplatnění u celých populací na celostátní úrovni.

IV. POPIS UPLATNĚNÍ CERTIFIKOVANÉ METODIKY

Tato metodika může být využita při výzkumných šetřeních souvisejících s vyhodnocováním genetických markerů a jejich vztahů ku užitkovým vlastnostem. Hlavním smyslem metodiky je uplatnění popsaných postupů při úpravách počítačových programů pro pravidelné celostátní předpovědi plemenných hodnot u všech hodnocených vlastností dojeného skotu v ČR. Metodika bude uplatněna ČMSCH a.s., která je ze zákona pověřena celostátním genetickým hodnocením hospodářských zvířat. ČMSCH koná tyto činnosti jako státní zakázku.

V. EKONOMICKÉ ASPEKTY

Podle zákona č. 110/1997 Sb. O potravinách a zákona č. 154/2000 Sb. O šlechtění, plemnitbě a evidenci hospodářských zvířat ve znění pozdějších předpisů je ČMSCH právnická osoba pověřená ministerstvem k výkonu činností podle jednotlivých bodů § 23c. Jmenovitě podle odstavců 1 a 2 a §7 je povinna poskytovat chovatelům a oprávněným osobám údaje, zpracovávat, zveřejňovat a evidovat výsledky. V souladu s doporučením Rady vlády pro výzkum uvádíme, že ČMSCH nevytváří těmito činnostmi zisk, poskytuje široké chovatelské veřejnosti co nejobjektivnější údaje a vyhodnocením celostátních databází vytváří podklady pro prokázání kvality plemnářské práce chovatelů.

Zavedení předpovědi GEPH je z hlediska celosvětového vývoje nutné. Celosvětově se přechází na stále novější postupy hodnocení zvířat. Tato nová domácí hodnocení se předávají do pravidelných celosvětových zpracování prováděných Interbullem. Interbull pravidelně „validuje“ postupy národních hodnocení, aby byla celosvětově porovnatelná a nenarušovala mezinárodní srovnání. Je nutno přizpůsobit domácí hodnocení tomuto mezinárodnímu vývoji. Změna dosavadních postupů podle předpokládané metodiky je nezbytná pro budoucí zapojení v mezinárodním hodnocení. Přínos metodiky spočívá v úpravě postupů práce celostátní infrastruktury. Peněžní vyjádření nelze tedy

stanovit, ale při nezavedení metodiky bude ČR z mezinárodních hodnocení vyřazena a plemenný materiál nebude prodejný. Postupy zpracování celostátních souborů jsou upravovány trvale. Jednokroková metoda má podstatně nižší nároky na úpravu počítačových programů pro práci s celostátními soubory než ostatní postupy, čímž dochází k šetření pracovních sil i peněžních prostředků.

Výsledky nových hodnocení budou využívat na celostátní úrovni, jak chovatelé, tak plemenářské podniky a státní správa.

VI. SEZNAM POUŽITÉ SOUVISEJÍCÍ LITERATURY

- Aguilar, I., Misztal, I. 2008. Technical note: Recursive algorithm for inbreeding coefficients assuming nonzero inbreeding of unknown parents. *J. Dairy Sci.*, 91: 1669-1672.
- Aguilar, I., Misztal, I., Johnson, D.L., Legarra, A., Tsuruta, S., Lawlor, T.J. 2010a. Hot topic: A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. *J. Dairy Sci.*, 93: 743-752.
- Aguilar, I., Misztal, I., Legarra, A., Tsuruta, S. 2010b. Efficient computation of genomic relationship matrix and other matrices used in the single-step evaluation. 9th WCGALP, Lipsko, Německo, 1.-6.8.2010.
- Bömcke, E., Gengler, N. 2009. Combining microsatellite and pedigree data to estimate relationships among Skyros ponies. *J. Appl. Genet.*, 50: 133-143.
- Bömcke, E., Soyeurt, H., Szydlowski, M., Gengler, N. 2009. How to combine pedigree and marker information into a single estimator for the calculation of relationships? EAAP-60th Ann. Meet., Barcelona, Španělsko.
- Ducrocq, V. 2011. Where are we heading to with genomics in dairy cattle evaluations? Interbull Technical Workshop: Establishing the framework for International Genomic Evaluations. Guelph, Ontario, Canada, 27.-28.2.2011.
- Forni, S., Aguilar, I., Misztal, I. 2011. Different genomic relationship matrices for single-step analysis using phenotypic, pedigree and genomic information. *Genet. Sel. Evol.*, 43: 1
- Guo, S.W. 1996. Variation in genetic identity among relatives. *Hum. Hered.*, 46: 61-70.
- Hayes, B.J., Bowman, P.J., Chamberlain, A.J., Goddard, M.E. 2009. Invited review: Genomic selection in dairy cattle: progress and challenges. *J. Dairy Sci.*, 92: 433-443.
- Henderson, C.R. 1975. Rapid method for computing the inverse of a relationship matrix. *J. Dairy Sci.*, 58: 1727-1730.
- Chen, C.Y., Misztal, I., Aguilar, I., Tsuruta, S., Meuwissen, T.H.E., Aggrey, S.E., Wing, T., Muir, W.M. 2011. Genome-wide marker-assisted selection combining all pedigree phenotypic information with genotypic data in one step: An example using broiler chickens. *J. Anim. Sci.*, 89: 23-28.
- Christensen, O.F., Lund, M.S. 2010. Genomic prediction when some animals are not genotyped. *Genet. Sel. Evol.*, 42:2.
- Jakubec, V., Říha, J., Golda, J., Majzlík, I. 1999. Odhad plemenné hodnoty hospodářských zvířat. VÚCHS Rapotín. 175 str.
- Madsen, P. 2010. DMU - Genomics. 6 str.
- Madsen, P., Jensen, J. 2008. DMU - A package for analysing multivariate mixed models. Version 6, release 4.7. 33 str. (<http://dmu.agrsci.dk>)

- Madsen, P., Su, G., Labouriau, R., Christensen, O.F. 2010. DMU - A package for analysing multivariate mixed models. 9th WCGALP, Lipsko, Německo, 1.-6.8.2010.
- Meuwissen, T.H.E., Hayes, B.J., Goddard, M.E. (2001): Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157: 1819-1829.
- Misztal, I. 2011. Osobní sdělení. 23.2.2011.
- Misztal, I. Gianola, D. 1987. Indirect solution of mixed model-equation. *J. Dairy Sci.*, 70: 716-723.
- Misztal, I, Legarra, A, Aguilar, I. 2009. Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. *J. Dairy Sci.*, 92: 4648-4655.
- Misztal, I. Aguilar, I., Legarra, A., Tsuruta, S., Johnson, D.L., Lawlor T.J. 2010. A Unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation. 9th WCGALP, Lipsko, Německo, 1.-6.8.2010.
- Misztal, I., Tsuruta, S., Strabel, T., Auvray, B., Druet, T., Lee, D.H., 2002. BLUPF90 and related programs (BGF90). 7th WCGALP, 19.–23.8.2002, Montpellier, France.
- Mrode, R.A. 2005. Linear models for the prediction of animal breeding values. 2nd ed., CABI Publishing, Wallingford, Oxfordshire OX10 8DE, UK. 344 str.
- Plemdat. 2011. www.plemdat.cz, 23.3.2011.
- Poschadel, N., Mayer, M. 2011. Realisierte Verwandtschaftsmatrizen und Zuchtwerschätzung für die genomische Selektion - eine Übersicht. *Züchtungskunde*, 83: 167-181.
- Quaas, R.L. 1976. Computing the diagonal elements and inverse of a large numerator relationship matrix. *Biometrics*, 32: 949-953.
- Schaeffer, L.R. 2006. Strategy for applying genome-wide selection in dairy cattle. *J. Anim. Breed. Genet.*, 123: 218-223.
- Szyda, J., Ptak, E., Komisarek, J., Zarnecki, A. 2008. Practical application of daughter yield deviations in dairy cattle breeding. *J. Applied Genetics*, 49: 183-191.
- Tsuruta S., Aguilar, I., Misztal, I., Legarra, A., Lawlor, T.J. 2010. Multiple trait genetic evaluation of linear type traits using genomic and phenotypic data in US Holsteins. 9th WCGALP, Lipsko, Německo, 1.-6.8.2010.
- VanRaden, P.M. 2008. Efficient methods to compute genomic predictions. *J. Dairy Sci.*, 91: 4414-4423.
- VanRaden P.M., Van Tassell C.P., Wiggans G.R., Sonstegard T.S., Schnabel R.D., Taylor J.F., Schenkel, F.S. 2009. Invited review: Reliability of genomic predictions for North American Holstein bulls. *J. Dairy Sci.*, 92: 16–24.
- VanRaden, P.M., Wiggans, G.R. 1991. Derivation, calculation, and use of national animal model information. *J. Dairy Sci.*, 74: 2737–2746.
- Verbyla, K.L., Calus, M.P.L., Mulder, H.A., de Haas, Y., Veerkamp, R.F. 2010. Predicting energy balance for dairy cows using high-density single nucleotide polymorphism information. *J. Dairy Sci.*, 93: 2757-2764.

VII. SEZNAM PUBLIKACÍ, KTERÉ PŘEDCHÁZELY METODICE

- Příbyl, J. 1995. Využití markerů při selekci hospodářských zvířat (A way of using markers for farm animals selection) /in Czech/. *Živočiš. Vyr.*, 40: 375-382.
- Příbyl, J., Příbylová, J. 2009. Genomická plemenná hodnota a šlechtění dojeného skotu. *Náš chov*. 69., č.10: 24-28.
- Příbyl, J., Příbylová, J. 2002. Výběr vhodného modelu při vyhodnocování souboru údajů. XV. letní škola biometrie „Biometrické metody a modely v současné vědě a výzkumu“. Lednice na Moravě, 2.- 6.9. Sborník referátů. 41-50. ÚKZÚZ v Brně.
- Příbyl, J., Řehout, V., Čítek, J., Příbylová, J. 2010a. Genetic evaluation of dairy cattle using a simple heritable genetic ground. *J. Sci. Food Agric.*, 90: 1765-1773.
- Příbyl, J., Řehout, V., Čítek, J., Příbylová, J. 2010b. Procedures of evaluation of genomic breeding value. XXIV. Genetické dny, MZLU Brno 1.-3.9., Book of abstracts p. 31.
- Příbyl, J., Řehout, V., Čítek, J., Příbylová, J. 2010c. Ways of animal evaluation according the large number of SNP in connection with production traits. XIX. letní škola biometrie, Lednice na Moravě 6.-10.9. (v tisku)

Příloha 1

Program v SASu na stanovení a porovnání GEPH maticovým výpočtem

```
/*.....          gebvmat.sas          ..... */
/*          jednokroková GEPH příklad v IML
h2 = 0.25 ;          radek          158
lamb = 0.8 ;          radek          205
lokusu = 20 ;          radek          71,146,150
způsob vypočtu prumeru alely radek          151
pocety urovni efektu          radek          43,60,189,191,232,234
pocet radku tisku poradi          radek          388,405,424
byku v zakladne          radek          392
    porovnani podle skupin zvirat
        PH a GEPH, prumery, rozdily, korelace, poradi
    porovnani A H, A22 G, prumery, rozdily, korelace
.....7.3.2011 VUZV Josef Pribyl ..*/
options /*nocenter*/ ls=92 nodate ps=32767 nonumber ;
dm output 'clear';
dm log 'clear';

                                           /*..cte..*/
filename uz          'c:\j\pp\genomika/priklzav\uzit';
filename rod          'c:\j\pp\genomika/priklzav\rod';
filename genot          'c:\j\pp\genomika/priklzav\genot';
filename sezkr          'c:\j\pp\genomika/priklzav\sezkrav';
filename sezby          'c:\j\pp\genomika/priklzav\sezbyc';
filename sezge          'c:\j\pp\genomika/priklzav\sezgenot';
filename dmul          'c:\j\pp\genomika/priklzav\vysdmul';

                                           /*..pise..*/
filename roddmu          'c:\j\pp\genomika/priklzav\roddmu';
filename uz2          'c:\j\pp\genomika/priklzav\uzdmu';
filename aa          'c:\j\pp\genomika/priklzav\aa';
filename hh          'c:\j\pp\genomika/priklzav\hh';
filename a22          'c:\j\pp\genomika/priklzav\a22';
filename gg          'c:\j\pp\genomika/priklzav\gg';
filename phkk          'c:\j\pp\genomika/priklzav\phkk';
filename phbb          'c:\j\pp\genomika/priklzav\phbb';
filename phgg          'c:\j\pp\genomika/priklzav\phgg';
/*.....cteni uzitkovosti.....*/
data uzit;          title "nacteny soubor uzitkovosti";
infile uz; input ml j s vek sp;
    vek = vek - 27;          /* vek oprava na 27 mes */
    vek2 = vek*vek;
    sp = sp - 90;          /* SP oprava na 90 dnu */
    sp2 = sp*sp;
file uz2 ;
put s j ml vek vek2 sp sp2;
proc means;
    /*proc print;*/
/* .....tvorba matic .....*/
data yy; set uzit;          title "vektor uzitkovosti Y";
    keep uz ;
    uz = ml;
```

```

proc means; /*proc print;*/
data xxm; set užit; title "matice stad (SRO) X";
keep s1 - s8;
array xx s1 - s8;
do i = 1 to 8;
xx[i] = 0 ;
end;
do i = 1 to 8 ;
if s = i then xx[i] = 1 ;
end;
proc means; /*proc print;*/
data xreg; set užit; title "matice X pro regrese";
keep vek vek2 sp sp2;
proc means; /*proc print;*/
data zzm; set užit; title "matice jedincu Z";
keep j1 - j80 ;
array zz j1 - j80;
do i = 1 to 80;
zz[i] = 0 ;
end;
do i = 1 to 80 ;
if j = i then zz[i] = 1 ;
end;
proc means; /*proc print;*/run;
/*.....cteni matic z disku .....*/
data genot; title "nactene genotypy";
infile genot; input j g1 - g20; /* 20 SNP */
proc means; run;
data kravy; title " seznam krav";
infile sezkr; input kra;
proc means; run;
data byci; title " seznam byku";
infile sezby; input byc;
proc means; run;
data sezge; title "seznam genotypovanych jedincu";
infile sezge; input jge;
proc means; run;
data pribuz; title "nactene puvody";
infile rod; input j o m ;
file roddmu ; put j o m j ;
proc means; run;
/*.....*/
proc iml; /* matice pribuznosti */
use pribuz;
read all into b;
close pribuz;
n = nrow(b); /* pocet jedincu pribuznost */
/*Vytvorenie spodnej triangularnej matice L;
L=i(n);
do i=1 to n; *Zacni diagonalnym prvkom jedinca 1;
o = B[i,2]; m = B[i,3];
if o & m = 0 then L[i,i] = 1;
if o & m > 0 then do;
x = L[o,1:o]; x = x#x;

```

```

        a = (sum(x))*0.25;
        y = L[m,1:m]; y = y#y;
        c = (sum(y))*0.25;
        L[i,i] = sqrt((1 - a - c));
    end;
    else if o > 0 then do;
        x = L[o,1:o]; x = x#x;
        a = (sum(x))*0.25;
        L[i,i] = sqrt((1-a));
    end;
    else if m > 0 then do;
        y = L[m,1:m]; y = y#y;
        c = (sum(y))*0.25;
        L[i,i] = sqrt((1-c));
    end;
/*Pokracuj v danom stlpci s druhym jedincom
    a pocitaj mimodiagonalny prvok danho jedinca L[j,i];*/
do j=i+1 to n;
    o = B[j,2]; m = B[j,3];
    if o & m = 0 then L[j,i] = 0;
    if o & m > 0 then L[j,i] = 0.5*(L[o,i] + L[m,i]);
    else if o > 0 then L[j,i] = 0.5*(L[o,i]);
    else if m > 0 then L[j,i] = 0.5*(L[m,i]);
end;
end;

                                *Transponovanie matice L;
G=L`;
*vypocet matice A - matica pribuznosti sposob vypoctu podla Quass 1976,;
A = L*G;                                                                title
"matice A";

                                /*print a;*/
                                /* pribuznost A22 genotypovanych - vyber z A*/
use sezge;
    read all into sg;
    close sezge;
ng = nrow(sg);
a22 = j(ng,ng,0);                                                       title "matice A22";
do i = 1 to ng;
    f = sg[i];
    do j = 1 to ng;
        d = sg[j];
        a22[i,j] = a[f,d];
    end;
end;

                                print a22;                                title "matice G ";
use genot;
    read all into gt;
    close genot;
    t = gt[,2:21];
jedna = j(1,ng,1);
souce = jedna * t;
pruce = souce /((2*ng);        /*prumerna cetnost druhe alely */
q = j(ng,20,1);
/*q = 2#pruce # q;*/        /* prumer podle genotypovane populace*/

```



```

        q = q;                                /* prumer natvrdo 0.5 */
        tq = t - q;                            /*print pruce; print q; print t; print tq;*/
g = tq*tq` ;
        jm = trace(g)/ng;
gg = g/jm ;                                  /*print g; print jm;*/print gg;
/*.....reseni AM .....*/
h2 = 0.25;                                    /* sestaveni soustavy rovnic */
        k = (1-h2)/h2;
use yy;
        read all into y;
        close yy;
use xxm;
        read all into x1;
        close xxm;
use xreg;
        read all into x2;
        close xreg;
        x = x1||x2;
use zzm;
        read all into z;
        close zzm;
xy = x`*y;
zy = z`*y;
ps = xy//zy;
        xx = x`*x;
        xz = x`*z;
        zx = xz`;
        zz = z`*z;
        zza = zz + k*inv(a);
        hr = xx||xz;
        dr = zx||zza;
        ls = hr//dr;
                                                                    title "reseni soustavy AM, matice A";
c = inv(ls);
res = c*ps;                                  /*print res ;*/
        dia = vecdiag(a);
        dic = vecdiag(c);
        dic = dic[13:92];
        r2 = dia -k*dic/dia;
stada = res[1:8];                            /* rozklad reseni na useky */
regvek = res[9:10];
regsp = res[11:12];
                                                                    /*print stada regvek regsp ;*/
phr = res[13:92];
ph = phr||r2;
phk1 = ph[11:20,]; phk2 = ph[22:50,]; phk3 = ph[61:80,];
        phk = phk1//phk2//phk3;
phb1 = ph[1:10,]; phb2 = ph[21,];
        phb = phb1//phb2;
phg1 = ph[2,]; phg2 = ph[5,]; phg3 = ph[21,]; phg4 = ph[51:60,];
        phg = phg1//phg2//phg3//phg4;
                                                                    /*print phk phb phg ;*/
/*..... GEPH lamb = 0,8 dve moznosti, jako lamb, nebo do ggg .....*/
lamb = 1 ;                                    /* oprava pribuznosti */

```

```

ggg = 0.8*gg + 0.2*a22;          /* vazeni G A22, aby slo G invertovat */
gin = inv(ggg);
a22in = inv(a22);
    opra = lamb*(gin - a22in);
/* roztazeni matice genotypovanych zvirat na policka
    podle matice vseh jedincu */
op = j(n,n,0);
    do i = 1 to ng;
        f = sg[i];
        do j = 1 to ng;
            d = sg[j];
            op[f,d] = opra[i,j];
        end ;
    end ;
ain = inv(a);
    hin =ain + op ;          title "reseni soustavy AM, matice G do H";
zza = zz + k*hin;
    hr = xx||xz;
    dr = zx||zza;
    ls = hr//dr;
c = inv(ls);
res = c*ps;          /*print res ;*/
    h = inv(hin);
        /* spolehlivost nepracuje ???????? obtiz v H??*/
    dih = vecdiag(h); /* bezne stanoveni spolehlivosti na zaklade */
    dic = vecdiag(c); /* diagonalniho prvku inverze leve strany */
    dic = dic[13:92]; /*soustavy v tomto pripade nepracuje */
    r2 = dih -k*dic/*dih*/;
stadag8 = res[1:8];          /* rozklad reseni na useky */
regvekg8 = res[9:10];
regspg8 = res[11:12];
        /*print stadag regvekg regspg ;*/
phr = res[13:92];
ph = phr||r2;
phk1 = ph[11:20,]; phk2 = ph[22:50,]; phk3 = ph[61:80,];
    phkg8 = phk1//phk2//phk3;
phb1 = ph[1:10,]; phb2 = ph[21,];
    phbg8 = phb1//phb2;
phg1 = ph[2,]; phg2 = ph[5,]; phg3 = ph[21,]; phg4 = ph[51:60,];
    phggv8 = phg1//phg2//phg3//phg4;
        /*print phkg phbg phgg ;*/
/*..... GEPH lamb = 0,9 .....*/
lamb = 1 ;          /* oprava pribuznosti */
ggg = 0.9*gg + 0.1*a22;          /* vazeni G A22, aby slo G invertovat */
gin = inv(ggg);
a22in = inv(a22);
    opra = lamb*(gin - a22in);
/* roztazeni matice genotypovanych zvirat na policka
    podle matice vseh jedincu */
op = j(n,n,0);
    do i = 1 to ng;
        f = sg[i];
        do j = 1 to ng;
            d = sg[j];
            op[f,d] = opra[i,j];

```

```

                end ;
            end ;
ain = inv(a);
    hin =ain + op ;           title "reseni soustavy AM, matice G do H";
zza = zz + k*hin;
    hr = xx||xz;
    dr = zx||zza;
    ls = hr//dr;
c = inv(ls);
res = c*ps;                 /*print res ;*/
    h = inv(hin);
                                /* spolehlivost nepracuje ???????? obtiz v H??*/
    dih = vecdiag(h);
    dic = vecdiag(c);
    dic = dic[13:92];
    r2 = dih -k*dic/*dih*/;
stadag9 = res[1:8];          /* rozklad reseni na useky */
regvekg9 = res[9:10];
regspg9 = res[11:12];
                                /*print stadag regvekg regspg ;*/
phr = res[13:92];
ph = phr||r2;
phk1 = ph[11:20,]; phk2 = ph[22:50,]; phk3 = ph[61:80,];
    phkg9 = phk1//phk2//phk3;
phb1 = ph[1:10,]; phb2 = ph[21,];
    phbg9 = phb1//phb2;
phg1 = ph[2,]; phg2 = ph[5,]; phg3 = ph[21,]; phg4 = ph[51:60,];
    phggv9 = phg1//phg2//phg3//phg4;
                                /*print phkg phbg phgg ;*/
/*..... GEPH lamb = 1,0 .....*/
lamb = 1.0 ;                 /* oprava pribuznosti */
ggg = 0.999*gg + 0.001*a22; /* vazeni G A22, aby slo G invertovat */
gin = inv(ggg);
a22in = inv(a22);
    opra = lamb*(gin - a22in);
/* roztazeni matice genotypovanych zvirat na policka
    podle matice vseh jedincu */
op = j(n,n,0);
    do i = 1 to ng;
        f = sg[i];
        do j = 1 to ng;
            d = sg[j];
            op[f,d] = opra[i,j];
        end ;
    end ;
ain = inv(a);
    hin =ain + op ;           title "reseni soustavy AM, matice G do H";
zza = zz + k*hin;
    hr = xx||xz;
    dr = zx||zza;
    ls = hr//dr;
c = inv(ls);
res = c*ps;                 /*print res ;*/
    h = inv(hin);

```

```

/* spolehlivost nepracuje ???????? obtiz v H??*/
    dih = vecdiag(h);
    dic = vecdiag(c);
    dic = dic[13:92];
    r2 = dih -k*dic/*dih*/;
stadagl = res[1:8]; /* rozklad reseni na useky */
regvekg1 = res[9:10];
regspg1 = res[11:12];

/*print stadag regvekg regspg ;*/

phr = res[13:92];
ph = phr||r2;
phk1 = ph[11:20,]; phk2 = ph[22:50,]; phk3 = ph[61:80,];
    phkg1 = phk1//phk2//phk3;
phb1 = ph[1:10,]; phb2 = ph[21,];
    phbg1 = phb1//phb2;
phg1 = ph[2,]; phg2 = ph[5,]; phg3 = ph[21,]; phg4 = ph[51:60,];
    phggv1 = phg1//phg2//phg3//phg4;

/*print phkg phbg phgg ;*/

/*.....*/
                                title "tisk porovnani PH GEPH";
rozsta = stada - stadag8 ; /* rozdily stada PH a GEPH */
rozg89 = stadag8 - stadag9 ; /* rozdily geph podla lamb */
rozrvek = regvek - regvekg8 ; /* rozdily regrese vek */
rozreg89 = regvekg8 - regvekg9 ;
rozrsp = regsp - regspg8; /* rozdily regrese SP */
rozsp89 = regspg8 -regspg9 ;

    print stada stadag8 stadag9 stadagl rozsta rozg89 ;
    print regvek regvekg8 regvekg9 regvekg1 rozrvek rozreg89;
    print regsp regspg8 regspg9 regspg1 rozrsp rozsp89;

phkk = phk||phkg8||phkg9||phkg1;
    create phkkk from phkk;
    append from phkk;
phbb = phb||phbg8||phbg9||phbg1;
    create phbbb from phbb;
    append from phbb;
phgg = phg||phggv8||phggv9||phggv1;
    create phggg from phgg;
    append from phgg;

/*.....matice do sloupce .....*/
                                /* pribuznost A */

velsa = n*n;
sloa = j(velsa,3,0);
k = 1;
    do i = 1 to n;
        do j = 1 to n;
            sloa[k,1] = b[i,1];
            sloa[k,2] = b[j,1];
            sloa[k,3] = a[i,j];
            k = k + 1 ;
        end ;
    end ;
create aa from sloa;
append from sloa;

                                /* pribuznost H */

```

```

sloh = j(velsa,3,0);
k = 1;
  do i = 1 to n;
    do j = 1 to n;
      sloh[k,1] = b[i,1];
      sloh[k,2] = b[j,1];
      sloh[k,3] = h[i,j];
      k = k + 1 ;
    end ;
  end ;
create hh from sloh;
append from sloh;

/* pribuznost G */

velslg = ng*ng;
slog = j(velslg,3,0);
k = 1;
  do i = 1 to ng;
    do j = 1 to ng;
      slog[k,1] = sg[i];
      slog[k,2] = sg[j];
      slog[k,3] = gg[i,j];
      k = k + 1 ;
    end ;
  end ;
create maprge from slog;
append from slog;

/* pribuznost A22 */

sla22 = j(velslg,3,0);
k = 1;
  do i = 1 to ng;
    do j = 1 to ng;
      sla22[k,1] = sg[i];
      sla22[k,2] = sg[j];
      sla22[k,3] = a22[i,j];
      k = k + 1 ;
    end ;
  end ;
create mapra22 from sla22;
append from sla22;

/*..... zapis G, A22, A, H na disk .....*/
data aaa; set aa; /* zapis A */
  file aa; put coll - col3;
  j1 = coll; j2 = col2; pria = col3;
  keep j1 j2 pria;
data hhh; set hh; /* zapis H */
  file hh; put coll - col3;
  j1 = coll; j2 = col2; prih = col3;
  keep j1 j2 prih;
data ggg; set maprge; /* zapis G */
  if col2 < coll then delete ;
  file gg; put coll - col3;
  j1 = coll; j2 = col2; prig = col3;
  keep j1 j2 prig;
data a22; set mapra22; /* zapis A22 */

```

```

        if col2 < col1 then delete ;
file a22; put col1 - col3;
j1 = col1; j2 = col2; priA22 = col3;
keep j1 j2 priA22;
/*..... zapis PH na disk .....*/
data kph; set phkkk; /* kravy */
file phkk; put col1 - col8;
phk = col1; r2k = col2; gphk8 = col3; gr2k8 = col4; gphk9 = col5; gr2k9 =
col6; gphk1 = col7; gr2k1 = col8;
keep phk r2k gphk8 gr2k8 gphk9 gr2k9 gphk1 gr2k1;
data bph; set phbbb; /* byci */
file phbb; put col1 - col8;
phb = col1; r2b = col2; gphb8 = col3; gr2b8 = col4; gphb9 = col5; gr2b9 =
col6; gphb1 = col7; gr2b1 = col8;
keep phb r2b gphb8 gr2b8 gphb9 gr2b9 gphb1 gr2b1;
data gph; set phggg; /* genotypovani jedinci */
file phgg; put col1 - col6;
phg = col1; r2g = col2; gphg8 = col3; gr2g8 = col4; gphg9 = col5; gr2g9 =
col6; gphg1 = col7; gr2g1 = col8;
keep phg r2g gphg8 gr2g8 gphg9 gr2g9 gphg1 gr2g1;
run;
/*.....nacteni vysledku DMU .....*/
data dmul ;
title "DMU1";
infile dmul ; input a b c d kra e f dmul g ;
if a ne 4 then delete ;
keep kra dmul ;
proc means;
/*.....porovnani PH GEPH .....*/
data pork; title "porovnani PH GEPH krav";
set kph ;
rozphk = phk - gphk8 ; rozk89 = gphk8 - gphk9;
rozr2k = r2k - gr2k8 ;
abrophk = abs(rozphk); abro89 = abs(rozk89);
abror2k = abs(rozr2k);
/*proc means; var phk r2k gphk8 gr2k8 gphk9 gr2k9 gphk1 gr2k1
rozphk rozk89 rozr2k abrophk abro89 abror2k;
proc corr; var phk gphk8 gphk9 gphk1;
proc corr; var r2k gr2k8 gr2k9 gr2k1;*/
data kk8;
merge kravy pork; keep kra phk gphk8 gphk9 gphk1;
data kk8;
merge kk8(in=krava)dmul ; by kra; if krava ;
proc sort ; by descending gphk8;
data kk8;
set kk8; gporad8 = _n_ ;
proc sort ; by descending gphk9 ;
data kk9;
set kk8; gporad9 = _n_ ;
proc sort ; by descending dmul ;
data kdml;
set kk9; poraddml = _n_ ;
proc sort ; by descending phk ;
data kkkk;
set kdml; porad = _n_; /*if porad > 20 then delete;*/

```

```

proc print data = kkkk ;
proc corr; var phk gphk8 gphk9 gphk1 dmul ;
data porb; title "porovnani PH GEPH byku";
set bph ;
rozphb = phb - gphb8 ; rozb89 = gphb8 - gphb9;
rozr2b = r2b - gr2b8 ;
abrophb = abs(rozphb); abro89 = abs(rozb89);
abror2b = abs(rozr2b);
/*proc means; var phb r2b gphb8 gr2b8 gphb9 gr2b9 gphb1 gr2b1
rozphb rozb89 rozr2b abrophb abro89 abror2b;
proc corr; var phb gphb8 gphb9 gphb1;
proc corr; var r2b gr2b8 gr2b9 gr2b1; */
data bb8;
merge byci porb; keep byc phb gphb8 gphb9 gphb1;
data dml;
set dmul; byc = kra ; drop kra;
proc sort data = dml; by byc ;
data bb8;
merge bb8(in=byci) dml; by byc; if byci ;
proc sort ; by descending gphb8;
data bb8;
set bb8; gporad8 = _n_ ;
proc sort ; by descending gphb9 ;
data bb9;
set bb8; gporad9 = _n_ ;
proc sort ; by descending dmul ;
data bdml;
set bb9; poraddml = _n_ ;
proc sort ; by descending phb ;
data bbbb;
set bdml; porad = _n_; /*if porad > 20 then delete; */
proc print data = bbbb ;
proc corr; var phb gphb8 gphb9 gphb1 dmul;
data porz; title "porovnani PH GEPH byku v zakladne";
set porb ;
if _n_ > 10 then delete;
/*proc means; var phb r2b gphb8 gr2b8 gphb9 gr2b9 gphb1 gr2b1
rozphb rozb89 rozr2b abrophb abro89 abror2b;
proc corr; var phb gphb8 gphb9 gphb1;
proc corr; var r2b gr2b8 gr2b9 gr2b1; */
data bz8;
merge byci porz; keep byc phb gphb8 gphb9 gphb1;
if phb = . then delete ;
data dml;
set dmul; byc = kra ; drop kra;
proc sort data = dml; by byc ;
data bz8;
merge bz8(in=byci) dml; by byc; if byci ;
proc sort ; by descending gphb8;
data bz8;
set bz8; gporad8 = _n_ ;
proc sort ; by descending gphb9 ;
data bz9;
set bz8; gporad9 = _n_ ;

```

```

        proc sort ; by descending dmul ;
data bdm1;
set bz9; poraddml = _n_ ;
        proc sort ; by descending phb ;
data bbbz;
set bdm1; porad = _n_;
        if porad > 20 then delete;
        proc print data = bbbz ;
        proc corr; var phb gphb8 gphb9 gphb1 dmul;
data porg;          title "porovnani PH GEPH genotypovanych jedincu";
set gph ;
rozphg = phg - gphg8 ; rozg89 = gphg8 - gphg9;
rozr2g = r2g - gr2g8 ;
        abrophg = abs(rozphg); abro89 = abs(rozg89);
        abror2g = abs(rozr2g);
/*proc means; var phg r2g gphg8 gr2g8 gphg9 gr2g9 gphg1 gr2g1
        rozphg rozg89 rozr2g abrophg abro89 abror2g ;
proc corr; var phg gphg8 gphg9 gphg1;
proc corr; var r2g gr2g8 gr2g9 gr2g1;*/
data phge8;
merge sezge porg; keep jge phg gphg8 gphg9 gphg1;
data gmul;
        set dmul; jge = kra; drop kra;
proc sort data = gmul; by jge ;
data phge8;
merge phge8(in=genot) gmul; by jge; if genot ;
        proc sort ; by descending gphg8;
data phgeg;
set phge8; gporad8 = _n_ ;
        proc sort ; by descending gphg9 ;
data phge9;
set phge8; gporad9 = _n_ ;
        proc sort ; by descending dmul ;
data dmge1;
set phge9; poraddml = _n_ ;
        proc sort ; by descending phg ;
data phgegg;
set dmge1; porad = _n_;
        /*if porad > 20 then delete;*/
        proc print data = phgegg ;
        proc corr; var phg gphg8 gphg9 gphg1 dmul;
run;
/*.....porovnani pribuznosti .....*/
data dah;          title "porovnani uhlopricky A H ";
merge aaa hhh ;
        if j1 ne j2 then delete ;
        rozah = pria - prih ;
        abroah = abs(rozah);
        proc means; var pria prih rozah abroah ;
        proc corr; var pria prih ;
data ndah;          title "porovnani mimouhlopricky A H ";
merge aaa hhh ;
        if j1 = j2 then delete ;
        rozah = pria - prih ;

```



```

        abroah = abs(rozah);
    proc means; var pria prih rozah abroah;
    proc corr; var pria prih ;
data da22g;                                title "porovnaní uhloprický A22 G ";
merge a22 ggg ;
    if j1 ne j2 then delete ;
    roza22g = pria22 - prig ;
        abroa22g = abs(roza22g);
    proc means; var pria22 prig roza22g abroa22g ;
    proc corr; var pria22 prig ;
data nda22g;                                title "porovnaní mimouhloprický A22 G ";
merge a22 ggg ;
    if j1 = j2 then delete ;
    roza22g = pria22 - prig ;
        abroa22g = abs(roza22g);
    proc means; var pria22 prig roza22g abroa22g;
    proc corr; var pria22 prig ;
run;
/*.....konec.....*/

```

PŘÍLOHA 2

Úsek programu v SASu pro sestavení G

```
%let pocra = 526 ; /* pocet genotypovanych jedincu */
%let pocsl = 54609 ; /* pocet SN */
%let pocsl0 =45173228 ; /* pocet radku souboru SN */
data c ; title "cisla byku v snp od 1 " ;
    set c ;
    keep ncj co cissn sn ; /* ciselniky jedincu (nova cisla, stara cisla),
                                cislo SNP, hodnota SNP*/

proc means ;
proc sort ; by cissn ; /* cislo SNP */
data _null_ ; title "vypocet prumeru " ;
    keep cissn souc pocet ;
    file souct ;
    set c ; by cissn ;
        if first.cissn then do ;
            souc = 0 ; pocet = 0 ;
        end ;
        if sn ne . then pocet + 1 ;
        if sn ne . then souc + sn ;
        if last.cissn then put cissn pocet souc ;
    data prusn ; title "prumer a souhrnny geneticky rozptyl " ;
        infile souct ;
        input cissn pocet souc ;
            if pocet = . or pocet < 1 then prum = . ;
            else prum = souc/pocet ;
        rozpt = prum*(2-prum)/2 ;
        proc means ;
data pruc ; title "odecteny prumery od sn " ;
    merge c(in= ccel) prusn ; by cissn ;
    snpr = sn - prum ;
    keep ncj co cissn snpr ;
    proc means ;
run ;
/*.....do matic .....*/
proc iml;
t = j(&pocra, &pocsl, . ); /*prazdna matice SNP */
use pruc;
do i = 1 to &pocsl0 ; /* pocet radku celeho souboru SNP*/
    read next ;
    t[co,cissn] = snpr ;
end ;
close pruc;
do i = 1 to &pocra ;
do j = 1 to &pocsl ;
    if t[i,j] = . then t[i,j] = 5 ; /* za chybejici dosadit 5 */
end ; end ;
/*.....rozezsane nasobeni - pomale.....*/
```

```

* pocsn = j(&pocra, &pocra, 0); /* vynulovany citatel G*/
* kolik = j(&pocra, &pocra, 0); /*vynulovane pocty pripadu na policko*/
* do i = 1 to &pocra; /* rozepsane nasobeni t*t' */
* do j = i to &pocra;
* do k = 1 to &pocsl;
* if t[i,k] < 5 & t[j,k] < 5 then do;
* pocsn[i,j] = pocsn[i,j] + t[i,k]*t[j,k];
* kolik[i,j] = kolik[i,j] + 1;
* end;
* end; /* prepocet na plny pocet SNP*/
* pocsn[i,j] = pocsn[i,j]*&pocsl/kolik[i,j];
* pocsn[j,i] = pocsn[i,j];
* end;
* end;
/*.....totez maticovym nasobenim - rychlejsi.....*/
pt = j(&pocra, &pocsl, 0); /*pocty nasobeni pro jednotlivy prvky G*/
do i = 1 to &pocra;
do j = 1 to &pocsl;
if t[i,j] > 2 then do; /* za chybejici dosadit 0 */
t[i,j] = 0;
pt[i,j] = 0;
end;
else pt[i,j] = 1;
end; end;
pocsn = t*t(t);
kot = pt*t(pt);
do i = 1 to &pocra;
do j = 1 to &pocra;
kot[i,j] = 1/kot[i,j];
end; end;
pocsn = &pocsl*pocsn#kot;
/*.....*/ /* citatel G*/
jmg = trace(pocsn)/&pocra; print jmg; /* porovnat s rozptylem */
/* normovana G*/
G = pocsn/jmg;
do i = 1 to &pocra;
do j = 1 to &pocra;
if j < i then g[i,j] = .;
end;
end;
varnam = "i1" : "i&pocra "; /*G matici do souboru*/
create maprge from g [colname = varnam];
append from g;
/*.....zapis G sloupec do souboru .....*/
data ggg; set maprge; title "zapis G ve sloupci ";
co = _n_;
data cgg;
set ggg;
array i i1 - i&pocra ;
file gg;
do j = 1 to &pocra ;

```

```

        put co j i[j] ;
    end ;
data slog ;
    infile gg ;
        input co j2 g ;
    if j2 < co then delete ;
    proc sort ; by co ;
data sl1cis ;
    merge slog(in=matg) d ; by co ; if matg ;
    ncj1 = ncj ;
        co = j2 ;
        keep ncj1 co g ;
    proc sort ; by co ;
data sl2cis ;
    merge sl1cis(in=matg2) d ; by co ; if matg2 ;
    ncj2 = ncj ;
        keep ncj1 ncj2 g ;
    proc sort ; by ncj1 ncj2 ;
data zapg ;
    set sl2cis ;
        g = round(g, 0.0001);
    file slo4g ;
    put ncj1 ncj2 g ;

```

/* trojuhelnik G */

/* precislovani co na ncj (jedinec)*/

/* zapis trojuhelnik G na disk */

PŘÍLOHA 3

Úsek programu v SASu pro sestavení A_{22}

(zpracoval Ing. J. Pjontek, PhD.)

```

proc iml ;
use grod6;
    read all into mb;
    close grod6;
n = nrow(mb);
L:=j(n);
do i=1 to n;
    o = mBfi 2; m = mBfi 3;
    if o & m = 0 then L[i,i] = 1;
    if o & m > 0 then do;
        x = L[o,1:o]; x = x#x;
        a = (sum(x))*0.25;
        v = L[m,1:m]; v = y#y;
        c = (sum(v))*0.25;
        L[i,i] = sqrt((1 - a - c));
    end;
    else if o > 0 then do;
        x = L[o,1:o]; x = x#x;
        a = (sum(x))*0.25;
        L[i,i] = sqrt((1-a));
    end;
    else if m > 0 then do;
        v = L[m,1:m]; v = y#y;
        c = (sum(y))*0.25;
    end;
end;

```

/* rodokmen jedinec, otec, matka */

/* nocet iedincu pribuznost vsetne nredku */

*Vytvorenie spodnej triangularnej matice L;

*Zacni diagonalnym prvkom jedinca 1;

```

                                L[i,i] = sqrt((1-c));
                                end;
/*Pokracui v danom stlnci s druhvm iedincom
                                a pocitaj mimodiagonalny prvok danho jedinca L[j,i];*/
                                do i=i+1 to n
                                o = mB[i.2]; m = mB[i.3];
                                if o & m = 0 then L[i,i] = 0;
                                if o & m > 0 then L[i,i] = 0.5*(L[o,i] + L[m,i]);
                                else if o > 0 then L[i,i] = 0.5*(L[o,i]);
                                else if m > 0 then L[j,i] = 0.5*(L[m,i]);
                                end;
                                end;
                                *Transponovanie matice L;
                                mG=L';
                                *vvnocet matice A - matica pribuznosti sposob vvnocetu nodla Ouass (1976);
                                A = I.*mG;                                title "matice A22 ";
                                /*print a;*/
                                /* pribuznost A22 genotypovanvch - vvber z A*/
                                /* seznam genotypovanych jedincu*/
                                use sezge;
                                read all into sg;
                                close sezge;
                                ng = nrow(sg);
                                a22 = i(ng ng 0);
                                do i = 1 to ng;
                                f = sg[i];
                                do i = 1 to ng;
                                d = sg[i];
                                a22[i,j] = a[f,d];
                                end;
                                end;
                                end;

```

Příloha 4

Algoritmy načítání velkých souborů a průběžné tvorby součtů

(návrhy od SAS)

```

/*zpusob 1 - vstup setriden v SAS datasetu*/
data one;
input a $ b $ c;
cards;
a3 b2 6
a3 b1 7
a4 b1 8
a1 b1 1
a1 b2 2
a2 b1 3
a2 b2 4
a2 b3 5
;
run;
proc sort data=one;
by a;
run;
data two (keep=a sumc pocet );
set one;
by a;
if first.a then do; sumc=0;pocet=0; end;
pocet+1;
sumc+c;
if last.a then output;
run;

```

```

/*zpusob 2 - vstup setriden v txt souboru */
data three (rename=(last_a =a));
infile datalines eof=finito;
input a $ b $ c;
retain last_a ;
keep last_a sumc;
if last_a eq a then do;
    sumc+c;
end;
else do;
    if last_a ne "" then output;
    sumc=c;
    last_a=a;
end;
return;
finito:
do;
    output;
end;
cards;
a1 b1 1
a1 b2 2
a2 b1 3
a2 b2 4
a2 b3 5
a3 b2 6
a3 b1 7
a4 b1 8
;
run;

/*zpusob 3 - vstup nesetriden (pekelné rychlé)*/
data _null_;
infile datalines eof=finito;
input a $ b $ c ;
if _n_ eq 1 then do;
    dcl hash suma(ordered:'a');
    suma.definekey('a');
    suma.definedata('a','sumc');
    suma.definedone();
end;
rc=suma.find();
if rc eq 0 then do;
    sumc=sumc+c;
    rc=suma.replace();
end;
else do;
    sumc=c;
    rc=suma.add();
end;
return;
finito:
do;
    suma.output(dataset:'four');
end;
cards;
a3 b2 6
a3 b1 7
a4 b1 8
a1 b1 1
a1 b2 2
a2 b1 3
a2 b2 4
a2 b3 5
;
run;

```

PŘÍLOHA 5

Popis věty výstupu z laboratoře (část)

#	Field	Description
1	Sample ID	Unique identification code of the DNA
2	Sample Name	Unique identification code of the DNA
3	Sample Index	Row index indicating the position of the Sample Name in BeadStudio project
4	SNP Name	Unique identification code of the SNP
5	SNP Index	Row index indicating the position of the SNP Name ID in BeadStudio project
6	SNP	Nucleotide change of a SNP
7	Chr	Chromosome number of the SNP
8	Position	SNP position on chromosome
9	Allele1 - AB	Allele definition at the locus in a A vs B format (indicating homozygosity or heterozygosity)
10	Allele2 - AB	
11	Allele1 - Forward	Allele definition expected based on the comparison with reference sequence. This nomenclature is genome build-dependent
12	Allele2 - Forward	
13	Allele1 - Top	Illumina's allele definition that let you code the allele without ambiguity and changes due to the genome build you're referring to: 1. NOT-ambiguous SNPs, like A/C,G or T/C,G, Allele1 is always A or T, but A is Allele1-TOP and T is Allele1-BOT (Bottom). 2. In the case of ambiguous SNPs, like A/T and C/G Illumina's method goes reading -n-bases upstream and +n-bases downstream the SNP position up to an unambiguous situation. When such a condition is found, the nomenclature is referred to the NOT-ambiguous SNP calling (for more details see Annex 1)
14	Allele2 - Top	
15	X	Normalized intensity of the A allele, referred to the A vs B format
16	Y	Normalized intensity of the B allele, referred to the A vs B format
17	X Raw	Raw intensity of the A allele, referred to the A vs B format
18	Y Raw	Raw intensity of the B allele, referred to the A vs B format
19	R	Normalized R-value (also named NormR) of a SNP for the sample. R-value is the intensity of the fluorescence signal
20	Theta	Normalized Theta-value (also named NormTheta) of a SNP for the sample, where Theta represents the angle deviation from pure allele A signal. Theta = 0 means pure

BOVINE Project

7/9

/*.....konec.....*/

Vydal: Výzkumný ústav živočišné výroby, v.v.i.
Přátelství 815, 104 00 Praha Uhřetěves

Název: **Úprava soustav rovnic pro předpověď genomické plemenné hodnoty (GEPH)
jedenkrokovou metodou**

Autoři: prof. Ing. J. Příbyl, DrSc., Mgr. Ing. J. Bolečková, Mgr. J. Haman,
Ing. T. Kott, PhD., Ing. J. Příbylová, CSc., RNDr. M. Šimečková, Dr. nat. techn.,
Ing. L. Vostrý, PhD., Ing. L. Zavadilová, CSc.

ISBN: 978-80-7403-079-6

Zpracováno v rámci řešení výzkumného projektu NAZV č. QI111A167