A Supplementary data

A.1 Workflow

A.1.1 Liquid handling

Table 15: Test for dispensing accuracy of the piezo nanodispenser with aqueous solutions of a FAM labeled oligonucleotide in several concentrations. Each concentration (0.1 - 0.5 M) and volume (100 nl - 200 nl) was dispensed in 8 replicates.

	FAM concentration [µM]								
	0.1	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**
200 nl									
$Mean_{Intensity}$	4020	5142	6111	7509	9414	11244	13086	14796	16980
SD	275	330	334	500	153	280	246	226	299
CV	6.8%	6.4%	5.5%	6.7%	1.6%	2.5%	1.9%	1.5%	1.8%
150 nl									
$Mean_{Intensity}$	3021	3737	4428	5356	6556	7618	8762	9708	11049
SD	90	155	67	210	119	190	147	177	209
CV	3.0%	4.2%	1.5%	3.9%	1.8%	2.5%	1.7%	1.8%	1.9%
100 nl									
$Mean_{Intensity}$	3013	3601	4232	5125	6076	6989	8031	8828	10149
SD	114	147	154	175	191	261	257	236	209
CV	3.8%	4.1%	3.6%	3.4%	3.1%	3.7%	3.2%	2.7%	2.1%

A.2 qPCR based expression analysis

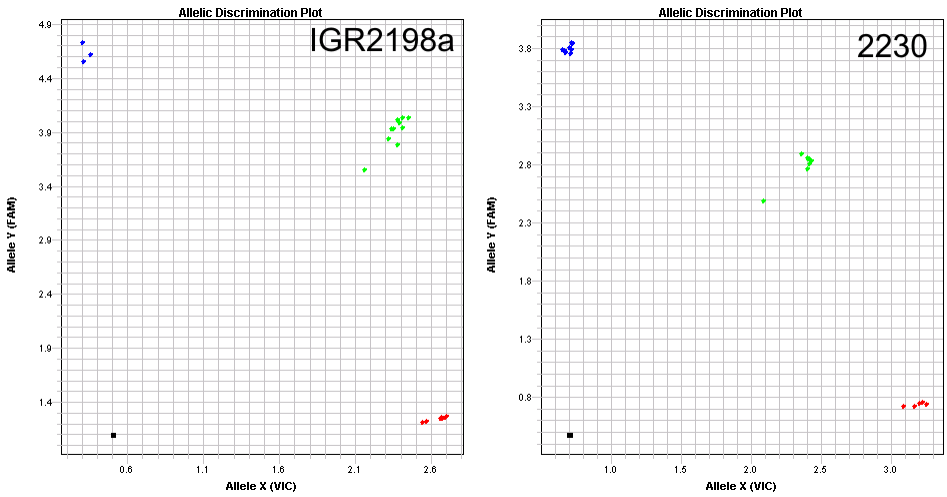
A.2.1 Comprehensive data sheet for mouse expression profiling (see next page)

Table 16: Mouse expression profiling, shown are absolute Ct-values and ΔCt-values after normalization against Actb.

	Brain				**Liver**				**Kidney**			
	200 nl		10 µl		200 nl		10 µl		200 nl		10 µl	
	Absolute Ct values											
	$\mathbf{N}_R$	**Ct±SD**	$\mathbf{N}_R$	**Ct±SD**	$\mathbf{N}_R$	**Ct±SD**	$\mathbf{N}_R$	**Ct±SD**	$\mathbf{N}_R$	**Ct±SD**	$\mathbf{N}_R$	**Ct±SD**
Cacna1g	6	25.7±0.17	4	24.5±0.14	6	-	4	-	6	31.5±1.49	4	30.2±0.31
Folh1	6	27.9±0.40	4	26.9±0.07	5	-	4	-	5	25.2±0.17	4	23.3±0.07
Odf2	6	25.9±0.27	4	25.1±0.05	5	30.1±0.5	4	28.7±0.07	6	27.6±0.18	4	26.8±0.09
Pttg1ip	5	25.9±0.23	4	25.0±0.06	5	25.3±0.1	4	23.5±0.05	6	23.8±0.10	4	22.2±0.18
Actb	6	21.7±0.26	4	19.3±0.04	6	23.9±0.9	4	20.8±0.06	6	21.4±0.36	4	18.9±0.02
	ΔΔCt-values against brain tissue after normalization (Actb)											
					ΔΔ*Ct*±SD		**ΔΔ*Ct*±SD**		**ΔΔ*Ct*±SD**		**ΔΔ*Ct*±SD**	
Cacna1g					-		-		-6.1±2.29		-6.2±0.51	
Folh1					-		-		2.4±1.19		3.3±0.20	
Odf2					-1.9±1.89		-2.1±0.22		-2.1±1.08		-2.1±0.20	
Pttg1ip					2.9±1.44		2.9±0.21		1.8±0.96		2.4±0.30	
	Fold changes to brain tissue											
Cacna1g					-		-		67.9		72.3	
Folh1					-		-		0.2		0.1	
Odf2					3.8		4.4		4.2		4.3	
Pttg1ip					0.1		0.1		0.3		0.2	

A.3 TaqMan based SNP genotyping

A.3.1 Examples of excellent allele calling in 10 µl



(a) Allele Calling for SNP IGR2198a in 10 µl

(b) Allele Calling for SNP 2230 in 10 µl

Figure 47: Visual representation of the allele calling for SNP IGR2198a and 2230 in 10 µl. Both SNPs show good clustering of all three genotypes.

A.3.2 Genotype information for samples provided by the University of Kiel

Center-Code	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b
	2011		2063		2230		3236		05IGR2096a		rs2272246	
37204	1	1	1	1	2	2	1	2	2	2	2	2
37198	1	2	1	2	1	2	1	2	1	2	2	2
37201	1	2	1	2	1	2	1	2	1	2	1	2
37282	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**2**	**2**
37280	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**
37278	**2**	**2**	**2**	**2**	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**1**
37334	**2**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**
37339	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**
37340	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
36748	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**
36761	**2**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**
36789	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**2**	**2**
36746	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
36790	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36745	**1**	**2**	**1**	**2**	**2**	**2**	**2**	**2**	**1**	**2**	**2**	**2**
36792	**1**	**2**	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**2**
36775	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**1**	**1**
36787	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36743	**2**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**
36788	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36837	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**1**	**2**
36763	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36791	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36783	1	1	1	2	1	1	1	1	1	2	2	2
36740	2	2	2	2	1	1	1	1	1	1	2	2
36741	1	2	1	2	1	2	1	1	1	2	2	2
36999	1	2	1	2	1	2	1	2	1	2	2	2
36998	0	0	0	0	0	0	0	0	0	0	0	0
37012	1	1	1	1	1	2	1	2	2	2	2	2
37011	1	2	1	2	1	2	1	2	1	2	1	2
37017	1	1	1	1	2	2	2	2	2	2	2	2
37016	1	1	1	1	2	2	1	2	2	2	2	2
37014	1	2	1	2	1	2	1	1	1	2	1	2
37001	1	2	1	2	1	2	1	2	1	2	2	2
37008	2	2	2	2	1	1	1	1	1	1	1	1
37004	1	2	1	2	1	2	1	2	1	2	2	2
36753	2	2	2	2	1	1	1	2	1	1	2	2
36738	1	1	1	1	2	2	2	2	2	2	1	2
36771	1	2	1	2	1	2	1	2	1	2	2	2
36779	1	2	1	2	1	2	1	2	1	2	1	2
36754	1	2	1	2	1	2	2	2	1	2	2	2
36757	1	1	1	1	2	2	2	2	2	2	1	2
36737	1	2	1	2	1	1	1	2	1	2	2	2
36945	1	1	1	1	1	2	1	2	2	2	2	2
36940	1	1	1	1	2	2	2	2	2	2	2	2
37003	2	2	2	2	1	1	1	1	1	1	2	2
37002	1	1	1	1	1	2	1	2	2	2	1	2
36703	1	1	1	2	1	2	1	2	1	2	1	1
36731	1	1	1	1	2	2	2	2	2	2	2	2
36734	1	2	1	2	1	2	1	1	1	2	1	2
36723	1	2	1	2	1	2	1	2	1	2	1	1
36711	1	1	1	2	1	2	1	2	1	2	2	2
36713	1	1	1	1	2	2	2	2	2	2	1	1
36716	1	1	1	1	2	2	1	2	2	2	2	2
36722	2	2	2	2	1	1	1	1	1	1	2	2
36785	0	0	0	0	0	0	0	0	0	0	2	2
36800	0	0	0	0	0	0	0	0	0	0	0	0
36693	1	2	1	2	1	2	1	2	1	2	2	2
36689	2	2	2	2	1	1	1	1	1	1	2	2
36673	1	1	1	2	1	2	2	2	1	2	2	2
36706	1	1	1	1	1	2	1	2	2	2	2	2
36617	1	2	1	2	1	2	1	2	1	2	2	2
36619	1	2	1	2	1	2	1	1	1	2	1	2
36629	1	1	1	2	1	1	1	1	1	2	1	2
36622	1	2	1	2	1	2	1	2	1	2	2	2
36627	1	1	1	2	1	2	2	2	1	2	1	2
36628	1	1	1	2	1	1	1	2	1	2	2	2
36642	1	2	1	2	1	2	1	2	1	2	1	2
36632	1	2	1	2	1	2	1	2	1	2	2	2
36633	1	2	2	2	1	1	1	1	1	1	1	2
36640	1	2	1	2	1	2	1	2	1	2	2	2
60784	1	1	1	1	2	2	1	2	2	2	1	2

Table 17 – SNP genotypes provided by University of Kiel

Center-Code	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b
	rs1264703		rs1045642		rs7195236		IGR2198a_1		rs1050152		rs2073838	
37204	2	2	1	1	1	2	1	1	1	1	1	2
37198	1	2	2	2	1	2	1	2	1	2	1	1
37201	1	2	1	2	1	2	1	2	1	2	1	1
37282	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
37280	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
37278	**1**	**1**	**1**	**2**	**1**	**1**	**2**	**2**	**2**	**2**	**1**	**1**
37334	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**1**
37339	**1**	**2**	**1**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
37340	**1**	**1**	**2**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
36748	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
36761	**1**	**2**	**1**	**2**	**1**	**1**	**2**	**2**	**2**	**2**	**1**	**1**
36789	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
36746	**1**	**1**	**2**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
36790	**1**	**2**	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
36745	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**	**1**	**1**
36792	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
36775	**1**	**1**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
36787	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**
36743	**1**	**1**	**2**	**2**	**1**	**1**	**2**	**2**	**2**	**2**	**1**	**1**
36788	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
36837	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**
36763	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
36791	**1**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
36783	1	2	1	2	1	1	2	2	2	2	1	1
36740	1	1	1	2	1	1	2	2	2	2	1	1
36741	1	1	1	2	1	1	1	2	1	2	1	1
36999	1	2	2	2	1	1	1	2	1	2	1	1
36998	0	0	0	0	0	0	0	0	0	0	0	0
37012	1	1	1	2	1	1	1	2	1	2	1	1
37011	1	1	1	2	1	1	1	2	1	2	1	1
37017	1	1	1	1	1	1	1	1	1	1	1	1
37016	1	1	2	2	1	2	1	1	1	1	1	1
37014	1	1	1	2	1	2	1	2	1	2	1	1
37001	1	2	1	2	1	1	1	2	1	2	1	1
37008	1	2	1	2	1	1	2	2	2	2	1	1
37004	1	2	1	2	1	2	1	2	1	2	1	1
36753	1	2	1	2	1	1	2	2	2	2	1	1
36738	1	2	1	2	1	1	1	1	1	1	1	1
36771	1	2	2	2	1	1	1	2	1	2	1	1
36779	1	1	0	0	1	1	1	2	1	2	1	1
36754	1	2	1	2	1	1	1	2	1	2	1	1
36757	1	2	1	2	1	2	1	1	1	1	2	2
36737	1	1	1	2	1	1	1	2	1	2	1	1
36945	1	1	2	2	1	2	1	1	1	1	1	1
36940	1	2	2	2	1	1	1	1	1	1	1	2
37003	1	1	1	1	1	1	2	2	2	2	1	1
37002	1	1	1	1	1	1	1	1	1	1	1	1
36703	1	2	1	2	1	1	1	2	1	2	1	2
36731	1	2	1	2	1	2	1	1	1	1	1	1
36734	1	1	1	2	1	1	1	2	1	2	1	1
36723	1	1	1	1	1	2	1	2	1	2	1	1
36711	1	1	1	2	1	2	1	2	1	2	1	1
36713	1	1	1	1	1	2	1	1	1	1	1	1
36716	1	2	1	1	1	2	1	1	1	1	1	1
36722	1	1	1	2	1	1	1	2	2	2	1	1
36785	1	2	1	2	0	0	0	0	0	0	0	0
36800	0	0	0	0	0	0	0	0	0	0	0	0
36693	1	2	1	2	1	1	1	2	1	2	1	1
36689	1	1	1	2	1	1	2	2	2	2	1	1
36673	1	1	1	1	1	1	1	2	1	2	1	1
36706	1	2	1	2	1	1	1	1	1	1	1	1
36617	1	2	1	2	1	1	1	2	1	2	1	1
36619	1	1	1	2	1	1	1	2	1	2	0	0
36629	1	2	1	2	1	1	1	2	1	2	1	1
36622	1	1	1	2	1	1	1	1	1	2	1	1
36627	1	1	1	1	1	2	1	2	1	2	1	1
36628	1	1	2	2	1	1	1	2	1	2	1	1
36642	1	1	1	2	1	2	1	2	1	2	1	1
36632	1	2	1	2	1	2	1	2	1	2	1	1
36633	1	1	1	2	1	1	2	2	2	2	1	1
36640	1	1	2	2	1	1	1	2	1	2	1	1
60784	1	1	1	1	1	1	1	1	1	1	1	2

Table 17 – continued from previous page

Center-Code	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b
	rs2631367		rs2862717		rs3087616		rs3822356		rs4279384		rs778584	
37204	1	1	2	2	2	2	1	1	2	2	1	1
37198	1	2	1	2	2	2	1	2	2	2	1	2
37201	1	2	2	2	2	2	1	2	2	2	1	2
37282	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
37280	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**
37278	**2**	**2**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**2**	**1**	**2**
37334	**2**	**2**	**1**	**2**	**2**	**2**	**1**	**2**	**2**	**2**	**1**	**2**
37339	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**1**
37340	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**2**
36748	**1**	**2**	**1**	**2**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36761	**2**	**2**	**1**	**2**	**2**	**2**	**2**	**2**	**1**	**2**	**1**	**2**
36789	**1**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**2**
36746	**1**	**2**	**2**	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**1**	**1**
36790	**1**	**1**	**1**	**1**	**2**	**2**	**1**	**1**	**1**	**2**	**1**	**1**
36745	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36792	**2**	**2**	**1**	**2**	**1**	**2**	**1**	**1**	**2**	**2**	**1**	**1**
36775	**1**	**1**	**2**	**2**	**2**	**2**	**1**	**2**	**2**	**2**	**1**	**2**
36787	**1**	**1**	**1**	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**1**	**1**
36743	**2**	**2**	**1**	**2**	**2**	**2**	**2**	**2**	**1**	**2**	**1**	**2**
36788	**1**	**1**	**2**	**2**	**2**	**2**	**1**	**2**	**2**	**2**	**1**	**2**
36837	**1**	**1**	**1**	**2**	**2**	**2**	**1**	**2**	**1**	**2**	**1**	**1**
36763	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**1**	**2**	**1**	**2**
36791	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
36783	2	2	2	2	2	2	1	1	2	2	1	1
36740	2	2	1	2	2	2	2	2	2	2	2	2
36741	2	2	1	2	1	2	2	2	1	1	1	1
36999	1	2	2	2	2	2	2	2	2	2	2	2
36998	0	0	0	0	0	0	0	0	0	0	0	0
37012	1	2	1	2	1	2	1	1	2	2	1	1
37011	1	2	1	2	1	2	1	1	2	2	1	1
37017	1	1	2	2	2	2	1	1	2	2	1	1
37016	1	1	1	2	1	2	2	2	2	2	2	2
37014	1	2	2	2	2	2	2	2	1	2	1	2
37001	1	2	2	2	2	2	2	2	1	2	1	2
37008	2	2	2	2	2	2	2	2	1	2	1	2
37004	1	2	1	1	1	2	1	2	2	2	1	2
36753	2	2	1	2	1	2	2	2	1	2	0	0
36738	1	2	1	2	2	2	1	1	2	2	1	2
36771	0	0	1	2	1	2	2	2	2	2	1	2
36779	1	2	0	0	2	2	2	2	2	2	2	2
36754	1	2	2	2	2	2	1	2	1	2	1	1
36757	1	1	2	2	2	2	1	2	2	2	1	2
36737	2	2	2	2	2	2	1	2	2	2	1	2
36945	1	2	2	2	2	2	2	2	1	1	1	2
36940	1	1	2	2	2	2	2	2	1	2	1	2
37003	2	2	2	2	2	2	1	2	2	2	1	2
37002	1	2	2	2	2	2	1	2	1	2	1	1
36703	1	2	1	2	1	2	1	2	2	2	1	2
36731	1	1	1	2	1	2	2	2	1	2	1	2
36734	1	2	1	2	1	2	2	2	1	2	1	2
36723	1	2	1	2	1	2	1	2	1	2	0	0
36711	1	2	2	2	2	2	1	2	2	2	1	2
36713	1	1	2	2	2	2	1	2	1	2	0	0
36716	1	1	1	2	2	2	1	2	1	2	1	1
36722	2	2	2	2	2	2	2	2	1	2	1	2
36785	0	0	0	0	2	2	0	0	0	0	0	0
36800	0	0	0	0	0	0	0	0	0	0	0	0
36693	1	2	1	2	1	2	2	2	1	1	1	1
36689	2	2	2	2	2	2	1	2	1	2	1	1
36673	1	2	2	2	2	2	2	2	2	2	2	2
36706	1	2	1	2	1	2	2	2	2	2	2	2
36617	1	2	1	2	1	2	1	2	2	2	1	2
36619	1	2	2	2	2	2	2	2	1	2	1	2
36629	2	2	2	2	2	2	1	2	2	2	1	2
36622	0	0	2	2	2	2	2	2	1	2	1	2
36627	1	2	2	2	2	2	2	2	1	2	1	2
36628	2	2	2	2	2	2	2	2	1	2	1	2
36642	1	2	2	2	2	2	2	2	1	2	1	2
36632	1	2	1	2	2	2	1	2	2	2	1	2
36633	2	2	2	2	2	2	2	2	2	2	2	2
36640	1	2	2	2	2	2	2	2	1	2	1	2
60784	1	1	1	1	1	2	2	2	1	2	1	2

Table 17 – continued from previous page

Center-Code	allele_a	allele_b	allele_a	allele_b	allele_a	allele_b
	rs9007		rs919744		X100	
37204	2	2	1	1	2	2
37198	1	1	1	1	1	2
37201	1	1	1	1	1	2
37282	**1**	**1**	**1**	**2**	**1**	**2**
37280	**1**	**1**	**2**	**2**	**1**	**2**
37278	**1**	**1**	**1**	**2**	**1**	**2**
37334	**1**	**2**	**1**	**1**	**1**	**1**
37339	**1**	**2**	**1**	**2**	**1**	**2**
37340	**1**	**1**	**1**	**2**	**1**	**2**
36748	**1**	**1**	**1**	**2**	**1**	**2**
36761	**1**	**1**	**1**	**1**	**1**	**1**
36789	**1**	**1**	**1**	**2**	**1**	**2**
36746	**1**	**1**	**1**	**1**	**1**	**2**
36790	**1**	**1**	**1**	**2**	**2**	**2**
36745	**1**	**1**	**1**	**1**	**2**	**2**
36792	**1**	**1**	**1**	**2**	**1**	**1**
36775	**1**	**1**	**1**	**1**	**2**	**2**
36787	**1**	**1**	**1**	**1**	**2**	**2**
36743	**1**	**2**	**1**	**1**	**1**	**1**
36788	**1**	**1**	**1**	**1**	**2**	**2**
36837	**1**	**1**	**1**	**1**	**2**	**2**
36763	**1**	**1**	**1**	**1**	**2**	**2**
36791	**1**	**2**	**1**	**1**	**2**	**2**
36783	1	2	1	1	1	1
36740	1	1	1	1	1	1
36741	1	1	1	2	1	2
36999	1	1	1	1	1	2
36998	0	0	0	0	0	0
37012	1	1	1	2	1	2
37011	1	1	1	2	1	2
37017	1	2	1	1	2	2
37016	1	1	1	2	1	2
37014	1	1	1	1	1	2
37001	1	1	1	1	2	2
37008	1	1	1	1	1	1
37004	1	1	1	2	1	2
36753	1	1	1	2	1	2
36738	1	1	1	1	2	2
36771	1	1	1	2	1	2
36779	1	1	1	1	1	2
36754	1	2	1	1	1	2
36757	1	2	1	1	2	2
36737	1	2	1	1	1	1
36945	1	1	1	1	1	2
36940	1	1	1	1	2	2
37003	1	1	1	1	1	1
37002	1	1	1	1	1	2
36703	1	1	1	2	1	2
36731	1	1	1	2	2	2
36734	1	1	1	2	1	2
36723	1	1	1	2	1	2
36711	1	1	1	1	1	2
36713	1	1	1	1	2	2
36716	1	1	1	1	2	2
36722	1	1	1	1	1	1
36785	1	1	0	0	0	0
36800	0	0	0	0	0	0
36693	1	1	1	2	1	2
36689	1	1	1	1	1	1
36673	1	2	1	1	2	2
36706	1	1	1	2	1	2
36617	1	1	1	2	1	2
36619	1	1	1	1	1	2
36629	1	1	1	1	1	1
36622	1	1	1	1	1	2
36627	1	1	1	1	1	2
36628	1	1	1	1	1	2
36642	1	1	1	1	1	2
36632	1	1	1	1	1	2
36633	1	1	1	1	1	1
36640	1	1	1	1	1	2
60784	1	1	1	2	0	0

Table 17: SNP genotypes provided by University of Kiel

A.3.3 Example of 200 nl SNP genotyping analysis summary

Table 18: After analysis data was assembled using Microsoft Excel to compare the generated data to the data from Kiel University.

Column VIC
Column FAM
Column VIC/FAM
Column FAIL
Patient info
Original Data from Kiel
Comparison Berlin vs. Kiel
rs3822356
CHIP1
CHIP2
Kiel
rs3822356

Centercodes	result					result					13			Chip1 vs. Kiel	Chip2 vs Kiel
36617	FAIL	0	0	1	1	HETERO	0	0	1	0	1	2	HETERO	FALSE	OK
36622	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36628	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36633	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36638	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36640	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36642	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36673	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36693	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36706	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36711	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36713	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36731	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36734	VIC	1	0	0	0	FAIL	0	0	1	1	2	2	VIC	OK	FALSE
36737	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36738	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36740	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36743	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36745	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36746	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36748	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36753	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36754	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36757	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36761	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36763	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36771	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36775	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36779	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36783	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36787	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36788	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36789	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36790	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36791	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36792	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
36800	HETERO	0	0	1	0	HETERO	0	0	1	0	0	0	VIC	FALSE	FALSE
36837	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
36940	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36945	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
36998	VIC	1	0	0	0	VIC	1	0	0	0	0	0	VIC	OK	OK
36999	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
37001	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
37002	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37003	FAM	0	1	0	0	HETERO	0	0	1	0	1	2	HETERO	FALSE	OK
37011	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
37012	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
37014	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
37016	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
37017	FAM	0	1	0	0	FAM	0	1	0	0	1	1	FAM	OK	OK
37198	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37278	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
37280	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37282	FAIL	0	0	1	1	VIC	1	0	0	0	2	2	VIC	FALSE	OK
37329	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37334	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37339	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
37340	HETERO	0	0	1	0	HETERO	0	0	1	0	1	2	HETERO	OK	OK
60784	VIC	1	0	0	0	VIC	1	0	0	0	2	2	VIC	OK	OK
NTC	FAIL	0	0	1	1	HETERO	0	0	1	0					
NTC	FAIL	0	0	1	1	FAIL	0	0	1	1					
NTC	FAIL	0	0	1	1	FAIL	0	0	1	1					

FAILING	3,4%	1,7%
MISSCLASSIFICATION	3,4%	1,7%
excluding 4 prom-samples	1,8%	0,0%

Summary

A.3.4 Comprehensive data from validation of allele calling algorithm

Table 19: In the left row (blue font) the genotypes are shown, which were obtained from large volumes. Misclassifications are marked in green. Failing reactions, which were determined from the VIC filter combination fluorescence intensities, are marked in yellow.

SNP1				SNP2				SNP3				SNP4				SNP5			
rs1050152	pos1	pos7		rs2073838	pos1	pos6	pos11	rs2631367	pos4	pos5		05IGR2096a	pos4	pos6	pos9	2011	pos2	pos3	
?	HETERO	HETERO		G/G	VIC	VIC	FAIL	?	FAIL	FAIL		T/G	HETERO	HETERO	FAIL	G/C	FAIL	FAIL	FAIL
?	VIC	VIC		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	VIC	VIC		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	HETERO	HETERO		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/G	HETERO	HETERO	FAIL	G/C	FAIL	FAIL	FAIL
?	HETERO	FAIL		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/G	HETERO	HETERO	FAIL	G/C	FAIL	FAIL	FAIL
?	HETERO	HETERO		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	FAM	FAM		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/T	VIC	VIC	FAIL	C/C	FAIL	FAIL	FAIL
?	FAM	FAM		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/T	VIC	VIC	FAIL	C/C	FAIL	FAIL	FAIL
?	VIC	VIC		G/A	VIC	HETERO	HETERO	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	HETERO	HETERO		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/G	HETERO	HETERO	FAIL	G/G	FAIL	FAIL	FAIL
?	VIC	VIC		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAIL	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	VIC	VIC		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	VIC	HETERO		A/A	FAM	FAM	FAM	?	FAIL	FAIL		G/G	HETERO	FAIL	FAIL	G/G	FAIL	FAIL	FAIL
?	VIC	VIC		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	HETERO	FAIL		G/G	VIC	FAIL	FAIL	?	FAIL	FAIL		T/G	FAIL	FAIL	FAIL	G/C	FAIL	FAIL	FAIL
?	HETERO	HETERO		G/G	VIC	VIC	VIC	?	FAIL	FAIL		G/G	FAM	FAM	FAIL	G/G	FAIL	FAIL	FAIL
?	FAM	FAM		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/T	VIC	VIC	FAIL	C/C	FAIL	FAIL	FAIL
?	HETERO	HETERO		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/G	HETERO	HETERO	FAIL	G/C	FAIL	FAIL	FAIL
?	HETERO	FAIL		G/G	VIC	HETERO	VIC	?	FAIL	FAIL		T/G	FAIL	FAIL	FAIL	G/C	FAIL	FAIL	FAIL
?	FAM	FAM		G/G	VIC	VIC	VIC	?	FAIL	FAIL		T/T	VIC	VIC	FAIL	C/C	FAIL	FAIL	FAIL

SNP6				SNP7				SNP8				SNP9				SNP10			
2063	pos2	pos7	pos12	2230	pos1	pos6	pos7	3236	pos4	pos5	pos10	IGR2198a	pos3	pos5	pos8	X100	pos2	pos3	pos8
C/G	HETERO	HETERO	HETERO	A/G	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	G/C	HETERO	HETERO	HETERO	A/G	HETERO	HETERO	HETERO
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAIL	FAM
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAM	FAM
C/G	HETERO	HETERO	HETERO	A/A	FAIL	FAIL	FAIL	A/A	VIC	VIC	VIC	G/C	HETERO	HETERO	HETERO	A/A	VIC	VIC	VIC
C/G	HETERO	HETERO	HETERO	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/C	HETERO	HETERO	HETERO	G/G	FAM	FAM	FAM
C/C	VIC	VIC	VIC	A/G	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	G/C	HETERO	HETERO	HETERO	A/G	FAM	HETERO	HETERO
G/G	FAM	FAM	FAM	A/A	FAIL	FAIL	FAIL	A/A	VIC	VIC	VIC	C/C	FAM	FAM	FAM	A/A	VIC	VIC	VIC
G/G	FAM	FAM	HETERO	A/A	FAIL	FAIL	FAIL	A/A	VIC	VIC	VIC	C/C	FAM	FAM	FAM	A/A	VIC	VIC	VIC
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAM	FAM
C/G	HETERO	HETERO	HETERO	A/G	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	G/C	HETERO	HETERO	HETERO	A/G	FAM	HETERO	HETERO
C/C	VIC	VIC	FAIL	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAM	FAIL
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAM	FAM
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAIL	G/G	FAIL	VIC	VIC	G/G	FAM	FAM	HETERO
C/C	VIC	VIC	VIC	G/G	FAIL	FAIL	FAIL	C/C	FAM	FAM	FAM	G/G	VIC	VIC	VIC	G/G	FAM	FAM	FAM
C/G	FAIL	VIC	FAIL	A/G	FAIL	FAIL	FAIL	A/C	FAIL	HETERO	VIC	G/C	FAIL	VIC	FAIL	A/G	FAIL	FAIL	FAIL
C/C	VIC	VIC	VIC	A/G	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	G/C	HETERO	HETERO	HETERO	A/G	FAM	HETERO	HETERO
G/G	FAM	FAM	FAM	A/A	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	C/C	FAM	FAM	FAM	A/G	FAM	HETERO	HETERO
C/G	HETERO	HETERO	HETERO	A/G	FAIL	FAIL	FAIL	A/C	HETERO	HETERO	HETERO	G/C	HETERO	HETERO	HETERO	A/G	FAM	HETERO	HETERO
C/G	HETERO	VIC	FAIL	A/G	FAIL	FAIL	FAIL	A/C	FAM	HETERO	HETERO	G/C	HETERO	HETERO	FAIL	A/G	FAM	HETERO	FAIL
G/G	FAM	FAM	FAM	A/A	FAIL	FAIL	FAIL	A/A	VIC	VIC	VIC	G/C	HETERO	HETERO	HETERO	A/A	VIC	VIC	VIC

A.3.5 Concordance rates SNP genotyping 60 patients vs. 16 SNPs

Table 20: Two identical µPCR chips of the 1k–format were processed. Targets are stated in the left column. Failings – shows the percentage of assays without amplification product. Concordance – shows the percentage of allele calling matching to the Kiel data. Concordance* – was introduced due to an accumulation of wrong allele calling for 4 samples in all targets, which is presumably caused by a DNA dispensing problem during the sample preloading or bad DNA quality, than by assays.

	hgDNA chip 1			**hgDNA chip 2**		
SNP-ID	Failings	Conc.	Conc.*	Failings	Conc.	Conc.*
rs1050152	1.7%	89.8%	96.4%	0.0%	93.2%	100.0%
rs2073838	1.7%	94.9%	98.2%	0.0%	96.6%	100.0%
05|GR2096a	3.4%	91.5%	96.4%	18.6%	96.6%	98.2%
2063	1.7%	93.2%	100.0%	0.0%	93.2%	100.0%
3236	1.7%	100.0%	100.0%	3.4%	100.0%	100.0%
IGR2198a	3.4%	93.2%	100.0%	0.0%	93.2%	100.0%
X100	6.8%	94.9%	98.2%	0.0%	93.2%	98.2%
rs4279384	8.5%	100.0%	100.0%	8.5%	86.4%	87.3%
rs3822356	3.4%	96.6%	98.2%	1.7%	98.3%	100.0%
rs778584	3.4%	94.9%	96.4%	0.0%	94.9%	96.4%
rs2862717	5.1%	100.0%	100.0%	1.7%	100.0%	100.0%
rs3087616	6.8%	96.6%	98.2%	6.8%	13.6%	12.7%
rs9007	5.1%	11.9%	12.7%	0.0%	91.5%	98.2%
rs919744	1.7%	93.2%	100.0%	1.7%	93.2%	100.0%
hCV7711033	3.4%	76.3%	76.4%	1.7%	94.9%	98.2%
rs4776116	8.5%	-	-	3.4%	-	-

A.4 Additional information about software used

A.4.1 Perl scripts

quant_scrip.pl to structure data in a spreadsheet like format

```
#!/usr/local/bin/perl

# asking for number of samples print ("number of samples: \n>>");
$cntE = <STDIN>; chomp($cntE);

# asking for the file with all intensity information 4 all pictures
print ("name of summary file: \n>>");
$file = <STDIN>;
chomp($file);

# asking for the name of the result file
print ("name of result file: \n>>");
$resultfile = <STDIN>;
chomp($resultfile);

# data file
$datei = $file;
open(datei, $datei) or die " Sorry, can't open file!";
@dateiliste = <datei>;
close (datei);

$res_datei = ">$resultfile";
open(res_datei, $res_datei) or die " Sorry, can't open file: " . $resultfile . "!";
for ($i=$cntA;$i<($cntE+1); $i++) {
    $cnt = $i . "n  ";
    print res_datei $cnt . "\t";
        foreach $zeile (@dateiliste) {
            if ($zeile=~ m/ $cnt/) {
                @setlist = split(/ * /,$zeile);
                $value = @setlist[5];
                $value =~s/ //;
                print res_datei $value . "\t";}

            }
        print res_datei "\n"; }
```

quanti_config.pl to combine intensities with a the plate configuration containing txt-file

```
#!/usr/local/bin/perl

# function for removing \n at the end of the line
sub trim {
    my @out = @_;
    for (@out) {
         s/^\s+//;
         s/\s+$//;
    }
    return wantarray ? @out : $out[0];
}

# asking for input file with values
print ("name of quantification file: \n>>");
$data_file = <STDIN>;
```

```
chomp($data_file);

# asking for the file with all intensity information for all pictures
print ("name of configuration  file:\n>>");
$config_file = <STDIN>;
chomp($config_file);

# asking for the name of the result file
print ("name of result file: \n>>");
$result_file = <STDIN>;
chomp($result_file);

# quantification data file
$datei = $data_file;
open(quanti_dat, $datei) or die " Sorry, can't open file!";
@data = <quanti_dat>;
close(quanti_dat);

# data file with configuration information
$datei1 = $config_file;
open(config_dat, $datei1) or die " Sorry, can't open file!";
# reading from file
while ( defined ($line = <config_dat>)) {
    push @data1, [ split(/\t/,$line) ];
 }
close(config_dat);


$res_datei = ">$result_file";
open(res_datei, $res_datei) or die " Sorry, can't open file: " . $result_file . "!";
foreach $zeile (@data) {
    @setlist = split(/\t/,$zeile);
    $key = $setlist[0];
    $key =~s/n//;
    if ($data1[$key][0] > 0) {
        $new_line = "$key\t" . $data1[$key][1] . "/" . $data1[$key][2] . "\t" . trim($data1[$key][3]);
        print res_datei $new_line;
        for ($a=1; $a<(scalar(@setlist));$a++) {
            print res_datei"\t$setlist[$a] ";}
    }
}
close(res_datei);
```

A.4.2 R statistics scripts

norm_nonlog4.R for qPCR analysis of 200 nl TaqMan assays

```
rm(list=ls()) # deleting of all old variables in workspace
graphics.off() # closes old graphical devices

# ******************** configuration parameters for script *********************************
# ***********************************************************************************************
# baseline area definition
lower.bl <- c(9) # lower baseline border (
upper.bl <- c(14) # upper baseline border
# name of input file
loc.file <- c("samba20050302_sum_quanti_fin_outlier_rm")
# name of experiment
chip_name <- "Samba20050302_outl_rm"
raw.threshold <- c(3.75) # threshold for ct calculation, based on logarithmic values of raw data
norm1.threshold <- 180 # threshold for ct calculation, based on normalised data (without SD)
# factor for interpolating each cycle, e.g. factor = 100 -> 15 cycles are
# splitted into 1500 datapoints for ct-value calculation
interpolfactor <- c(100)
export.file2 <- c(paste(chip_name,"_","bg_substracted_data",sep="")) # export BG substracted data
export.file3 <- c(paste(chip_name,"_","mean_cts",sep="")) # export mean ct values
export.file4 <- c(paste(chip_name,"_","all_cts",sep="")) # export all ct values
# ***********************************************************************************************

x.raw <- read.table(loc.file, strip.white = TRUE, row.names=1) # read txt-file as table
# variables for genelist and dilution factor (or something else)
gene <- as.character(x.raw[,1])
dil.fac <- as.character(x.raw[,2])
x.raw <- x.raw[,c(-1,-2)]
group <- strsplit(gene,"/")# grouping options
target <- lapply(group,function(x){x[1]}) # list with targets
target_tissue <- lapply(group,function(x){x[2]}) # list with tissues (or something else)
target <- unlist(target)
target_tissue <- unlist(target_tissue)
# change to matrix format, all values are getting numeric, which enables calculation
x.raw <- as.matrix(x.raw)
well_no <- row(x.raw)[,1]

# *************************** FUNCTIONS BLOCK ***********************************************
# ***********************************************************************************************
# function for generating grafical output into pdf-file
# data = data that shall be visualized (matrix format)
# typofdata = descriptive string naming the type of data like 'raw' or 'normalized'
bmp_plotting <- function(data,typeofdata,th=0,arg3=dil.fac,arg1=target,arg2=target_tissue,
                         pwidth=pic.width,pheight=pic.height)
{
for (single.arg1 in unique(arg1)) {
    # 1st layer of loop for each 'first argument'
    for (single.arg2 in unique(arg2)) {
        choice_arg <- (arg1==single.arg1)&(arg2==single.arg2)
        #
        data.bmp_plot1<- data[choice_arg,]
        #if (nrow(data.bmp_plot1) > 0)
        #   {
            # Plotten der Daten nach argument 1 und 2 zusammengefasst***
            plot(1:ncol(data.bmp_plot1),data.bmp_plot1[1,],t="n",ylim=range(data),
                 main = paste(typeofdata," for ",single.arg1,"(",single.arg2,")",sep=""),
                 xlab="Cycles",ylab="Intensities")
            for (i in 1:nrow(data.bmp_plot1)){
```

```
            lines(1:ncol(data.bmp_plot1),data.bmp_plot1[i,],col=i)
            }
        # plotting of averaged line
        for (single.arg3 in unique(arg3)) {
            data.bmp_plot1_sub <- data[(arg3==single.arg3)&(arg1==single.arg1)&(arg2==single.arg2),]
            data.bmp_plot1_sub <- data.bmp_plot1_sub[data.bmp_plot1_sub[,ncol(data.bmp_plot1_sub)]>th,]
            #print(dim(data.bmp_plot1_sub))
             if(length(data.bmp_plot1_sub) > ncol(data.bmp_plot1))
                {
                 data.bmp_plot1_sub  <- colMeans(data.bmp_plot1_sub)
                 lines(1:ncol(data.bmp_plot1),data.bmp_plot1_sub ,col=1,lty=2,lwd=2)
                }
            }
        abline(h=th,lty=3)
     #   }
     #else {break}
       }
    }
} # End of function bmp_plotting
# ********************************************************************************************
# function for generating grafical output into pdf-file
# data = data that shall be visualized (matrix format)
# typofdata = descriptive string naming te type of data like 'raw' or 'normalized'
# logarithmic Y-scale !!!!!!!!!!!!
bmp_plotting1 <- function(data,typeofdata,th=0,arg3=dil.fac,arg1=target,arg2=target_tissue,
                          pwidth=pic.width,pheight=pic.height)
{
for (single.arg1 in unique(arg1)) {
    # 1st layer of loop for each 'first argument'
    for (single.arg2 in unique(arg2)) {
        choice_arg <- (arg1==single.arg1)&(arg2==single.arg2)
        #
        data.bmp_plot1<- data[choice_arg,]
        #if (nrow(data.bmp_plot1) > 0)
        #   {
            # Plotten der Daten nach argument 1 und 2 zusammengefasst***
            plot(1:ncol(data.bmp_plot1),data.bmp_plot1[1,],t="n",ylim=range(0.5:max(data)),
                 main = paste(typeofdata," for ",single.arg1,"(",single.arg2,")",sep=""),
                 xlab="Cycles",ylab="Intensities", log = "y")
            for (i in 1:nrow(data.bmp_plot1)){
                lines(1:ncol(data.bmp_plot1),data.bmp_plot1[i,],col=i)
                }
            # plotting of averaged line
            for (single.arg3 in unique(arg3)) {
                data.bmp_plot1_sub <- data[(arg3==single.arg3)&(arg1==single.arg1)&(arg2==single.arg2),]
                data.bmp_plot1_sub <- data.bmp_plot1_sub[data.bmp_plot1_sub[,ncol(data.bmp_plot1_sub)]>th,]
                #print(dim(data.bmp_plot1_sub))
                 if(length(data.bmp_plot1_sub) > ncol(data.bmp_plot1))
                    {
                     data.bmp_plot1_sub  <- colMeans(data.bmp_plot1_sub)
                     lines(1:ncol(data.bmp_plot1),data.bmp_plot1_sub ,col=1,lty=2,lwd=2)
                    }
                }
            abline(h=th,lty=3)
        #   }
        #else {break}
       }
    }
} # End of function bmp_plotting
# ********************************************************************************************
```

```
# function for plotting of all ct-values versus well position to estimate variation
# function plots ct-values that are handed over against the well position(x-axis)
ct_dotplot <- function(data,arg3=dil.fac,arg1=target,arg2=target_tissue, median_dist=1)
# variables:
# data => array of ct-values
# arg3 => 3rd layer of the loop, here dilution factor
# arg2 => 2nd layer of the loop, here target tissue
# arg1 => 1st layer of the loop, here gene
# median_dist => length ob bars for the medain values, that show the acceptable variation range
{
plot(data,type = "n", main = "ct-Values for all Wells",xlab="well-no",ylab="ct-values")
color <- 0
pch1 <- 64
# 1st layer of loop for each 'first argument'
# draw grid in grey
abline(h = pretty(range(data,na.rm=T),dist(range(data,na.rm=T))), v = pretty(1:length(data),
      (length(data)/5)), col = "lightgray")
for (single.arg1 in unique(arg1))
    {
    # 2nd layer of loop for each 'second argument'
    for (single.arg2 in unique(arg2))
            {
            # 3rd layer for each third argument
            for (single.arg3 in unique(arg3))
                {
                choice <- (arg3==single.arg3)&(arg1==single.arg1)&(arg2==single.arg2)
                ct_subset <- data[choice]
                ct_well <- well_no[choice]
                color <- color + 1
                points(ct_well,ct_subset, pch=pch1, cex=0.5, col=color)
                points(mean(ct_well,na.rm=T),mean(ct_subset,na.rm=T),pch="*",cex=2,col=color)
                m_point <- median(ct_subset,na.rm=T) # median of ct-values
                points(mean(ct_well,na.rm=T),m_point,pch="+",cex=2,col=color)
                segments(mean(ct_well,na.rm=T),(m_point-median_dist),mean(ct_well,na.rm=T),
                (m_point+median_dist))
                if(color == 9){
                    color <- 0
                    pch1 <- pch1+1
                    }
                }
            }
    }
} # End of function
# ********************************************************************************************
# ****************************** End of Function Block ***************************************
# ********************************************************************************************

# calculation of baseline *** based on raw data ********************************************
baseline <- rowMeans(x.raw[,lower.bl:upper.bl]) # <- apply(x[,lower.bl:upper.bl],1,mean)
# BG-Substraction ('Normalization')
y1 <- t(scale(t(x.raw),center=baseline,scale=F))
# for export of data
y1_exp <- cbind(gene,dil.fac,y1)
# Extrapolation of Data for Ct-value calculation
# from end of baseline range to the last cycles has to be interpolated
auswahl <- upper.bl:ncol(x.raw)
# ** for normalized data ***************************************************************
# y1 equivalent to data matrix after the right baseline border
y1_norm <- y1[,auswahl]
# z1 => interpolated data
```

```
z1 <- matrix(nrow=nrow(y1_norm),ncol=ncol(y1_norm)*interpolfactor)
for (i in 1:nrow(y1_norm)){
    z1[i,] <- approx(auswahl,y1_norm[i,],n=ncol(y1_norm)*interpolfactor)$y
    }
colnames(z1) <- approx(auswahl,y1_norm[i,],n=ncol(y1_norm)*interpolfactor)$x
# ********************************************************************************************
# Calculation of Ct-values:
# ********************************************************************************************
zz1 <- z1>=norm1.threshold # setting of threshold
norm1.ct <- numeric(nrow(z1))
for (i in 1:nrow(z1)){
    norm1.ct[i] <- as.numeric(colnames(z1))[zz1[i,]][1]
}
# ********************************************************************************************
# Calculation of mean Ct-values:
# ********************************************************************************************
dil.fac1 <- as.vector(dil.fac)
dil.fac1 <- unique(dil.fac1)
ct.mean.list <-""
ct.mean.list <- as.numeric(ct.mean.list)
ct.mean.list1 <- c(paste("attribute_1","\t","attribute_2","\t","number of cucc. reaction",
                  "\t","mean-ct","\t","STDEV","\t","num. of reactions","\t",
                  "mean-ct-nonoutlier","\t","STDEV-nonoutlier",sep=""))
dummylist <- ""
for(each in unique(gene)){
    for(each1 in dil.fac1){
        choice.ct.mean <- norm1.ct[gene==each&dil.fac==each1]
        # calculation outlier free mean
        if(length(choice.ct.mean)>0)
            {
            choice.non_outlier <- (median(choice.ct.mean)+3*sd(choice.ct.mean,na.rm=T)) >
                                  choice.ct.mean & choice.ct.mean >
                                  (median(choice.ct.mean)-3*sd(choice.ct.mean,na.rm=T))
            choice.ct.mean_non_outlier <- choice.ct.mean[choice.non_outlier]
            dummylist <- append(dummylist,paste(each,"/",each1,":",choice.ct.mean_non_outlier,sep=" "))
            }
        else {choice.ct.mean_non_outlier <- "-"}
        # calculation of mean from all values
        ct.mean <- mean(choice.ct.mean, na.rm=T)
        if (length(choice.ct.mean) > 1) {ct.sd <- sd(choice.ct.mean,na.rm=T)}
        else { ct.sd <- "-"}
        # calculation from values without outliers
        ct.mean_non_outlier <- mean(choice.ct.mean_non_outlier, na.rm=T)
        if (length(choice.ct.mean_non_outlier) > 1) {ct.sd_non_outlier <-
            sd(choice.ct.mean_non_outlier,na.rm=T)}
        else { ct.sd_non_outlier <- "-"}
        ct.mean.list1 <- append(ct.mean.list1,paste(each,"\t",each1,"\t",length(choice.ct.mean),
                                "\t",ct.mean,"\t",ct.sd,"\t",length(choice.ct.mean_non_outlier),
                                "\t",ct.mean_non_outlier,"\t",ct.sd_non_outlier,sep=""))
        }
}
# ************************************ GENERATION OF GRAFICAL OUTPUT ********************
# ********************************************************************************************
file_name1 <- paste(chip_name,"_BGcorr_data",".pdf",sep="") # file for bg corrected data
file_name2 <- paste(chip_name,"_Raw_data",".pdf",sep="") # file name for raw data
file_name3 <- paste(chip_name,"_BGcorr_data_ID",".pdf",sep="") # file for bg corrected data
file_name4 <- paste(chip_name,"_Raw_data_ID",".pdf",sep="") # file name for raw data
# **** bg corrected data with logarithmic Y-axis ******************************************
file_name5 <- paste(chip_name,"_BGcorr_data_logscale",".pdf",sep="")
# *** BG corrected data in pdf file, grouped regarding the configuration scheme *************
```

```
pdf(file=file_name1)
bmp_plotting(y1,"BG Corrected Data",th=norm1.threshold)
dev.off()
# Raw data in pdf file, grouped regarding the configuration scheme *************************
pdf(file=file_name2)
bmp_plotting(x.raw,"RAW DATA")
dev.off()
# BG corrected data in pdf file, grouped regarding the configuration scheme, logarihmic Y-axis
pdf(file=file_name5)
bmp_plotting1(y1,"BG Corrected Data",th=norm1.threshold)
dev.off()
# defining number of windows for graphical output
par(mfrow=c(2,2),lab= c(length(x.raw[1,]),5,7))
# plotting of raw intensities ***************************************************************
plot(1:ncol(x.raw),x.raw[1,],t="n",ylim=range(x.raw),main = "Raw Data",xlab="Cycles",ylab="Intensities")
for (i in 1:nrow(x.raw)){
    lines(1:ncol(x.raw),x.raw[i,],col=i)
    }
abline(v=lower.bl)
abline(v=upper.bl)
# *******************************************************************************************
# plotting of background substracted data ***************************************************
plot(1:ncol(y1),y1[1,],t="n",ylim=range(y1),main = "BG Corrected Data ",xlab="Cycles",ylab="Intensities")
for (i in 1:nrow(y1)){
    lines(1:ncol(y1),y1[i,],col=i)
    }
abline(h=norm1.threshold,lty=3)
# plotting of Ct-values *********************************************************************
ct_dotplot(norm1.ct)
# plotting of bg-substracted data with logarithmic Y-axis ***********************************
plot(1:ncol(y1),y1[1,],t="n",ylim=range(0.5:max(y1)),main = "BG Corrected Data ",
     xlab="Cycles",ylab="Intensities ", log = "y")
for (i in 1:nrow(y1)){
    lines(1:ncol(y1),y1[i,],col=i)
    }
abline(h=norm1.threshold,lty=3)
# save screen plots in pdf file *************************************************************
# name of pdf-file
file_n <- paste(chip_name,"_summary",".pdf",sep="")
pdf(file=file_n)
# plotting instruction
plot(1:ncol(x.raw),x.raw[1,],t="n",ylim=range(x.raw),main = "Raw Data",xlab="Cycles",ylab="Intensities")
for (i in 1:nrow(x.raw)){
    lines(1:ncol(x.raw),x.raw[i,],col=i)
    }
abline(v=lower.bl)
abline(v=upper.bl)
# plotting of background substracted data, linear Y-axis ************************************
plot(1:ncol(y1),y1[1,],t="n",ylim=range(y1),main = "BG Corrected Data ",xlab="Cycles",ylab="Intensities")
for (i in 1:nrow(y1)){
    lines(1:ncol(y1),y1[i,],col=i)
    }
abline(h=norm1.threshold,lty=3)
# plotiing of background substracted data, logarithmic Y-axis*******************************
plot(1:ncol(y1),y1[1,],t="n",ylim=range(0.5:max(y1)),main = "BG Corrected Data ",
     xlab="Cycles",ylab="Intensities ", log = "y")
for (i in 1:nrow(y1)){
    lines(1:ncol(y1),y1[i,],col=i)
    }
abline(h=norm1.threshold,lty=3)
```

```
# plotting of Ct-values ***************************************************************
ct_dotplot(norm1.ct)
dev.off()
# ************************************** END OF GRAFICAL OUTPUT BLOCK ******************
# ***********************************************************************************************
# export of data
# export of bg-substracted data
write.table(y1_exp, file = export.file2, quote=F,sep="\t",dec=",")
# export of mean ct-values
write.table(ct.mean.list1, file = export.file3, quote=F,sep="\t",dec=",")
# export of ct-values  into txt.file
dataexport <- cbind(gene, dil.fac, norm1.ct )
write.table(dataexport, file  = export.file4, quote=F,sep="\t", dec=",")
```

allel_calling1.R for allele calling of 200 nl TaqMan assays

```
rm(list=ls())
graphics.off()

# configuration parameters for script
#################################
loc.file <- c("pos08_snp9.txt")
#location of data file
file_n <-paste(loc.file,"_summary",".pdf",sep="")
file_exp <- paste(loc.file,"_summary",".txt",sep="")

#################################
# read txt-file as table
x.raw <- read.table(loc.file, strip.white = TRUE)

# variables for FAMVIC, FAM, VIC and BG signals
FAMVIC <- as.numeric(x.raw[,1])
FAM <- as.numeric(x.raw[,2])
VIC <- as.numeric(x.raw[,3])
BG <- as.numeric(x.raw[,4])

INDEX <- FAMVIC - FAM - FAM + BG

# graphical output on screen
par(mfrow=c(2,2))

# plotting instruction
plot(INDEX,ylim=range(INDEX),main = "INDEX",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
VIC_threshold <- locator(n=1)[2] abline(h=VIC_threshold)
FAM_threshold <- locator(n=1)[2] abline(h=FAM_threshold)

# plotting instruction
plot(VIC,ylim=range(VIC),main = "VIC",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
VICline <- locator(n=1)[2]
abline(h=VICline)

# plotting instruction
plot(FAMVIC,ylim=range(FAMVIC),main ="FAMVIC",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
FAMVICline <- locator(n=1)[2]
abline(h=FAMVICline)

pdf(file=file_n)
# plotting instruction
plot(VIC,ylim=range(VIC),main = "VIC",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
abline(h=VICline)

# plotting instruction
plot(FAM,ylim=range(FAM),main = "FAM",xlab="Sample number",ylab="Intensities")
abline(v=20.5)

# plotting instruction
plot(FAMVIC,ylim=range(FAMVIC),main = "FAMVIC",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
abline(h=FAMVICline)
```

```
# plotting instruction
plot(INDEX,ylim=range(INDEX),main = "INDEX",xlab="Sample number",ylab="Intensities")
abline(v=20.5)
abline(h=VIC_threshold)
abline(h=FAM_threshold)

dev.off() # end of pdf output

# filtering of values
VIC_plus <- INDEX > VIC_threshold
FAM_plus <- INDEX < FAM_threshold
HETERO_plus <- INDEX < VIC_threshold & INDEX > FAM_threshold
FAMVIC_failure <- FAMVIC < FAMVICline
VIC_failure <- VIC < VICline

samples <- seq(1:length(FAM))

# assembly of result list
result <- ""
# final table generation
for(i in seq(1:length(FAM))){
    if(sum(VIC_plus[i],FAM_plus[i])==0) result[i] <- "HETERO"
    if(VIC_plus[i] > 0) result[i] <- "VIC"
    if(FAM_plus[i] > 0) result[i] <- "FAM"
    if(VIC_failure[i] > 0) result[i] <- "FAIL" }

# export into txt.file
dataexport <- cbind(samples, result, as.numeric(VIC_plus),  as.numeric(FAM_plus),
as.numeric(HETERO_plus),  as.numeric(FAMVIC_failure), as.numeric(VIC_failure) )
write(loc.file, file = file_exp)
write.table(dataexport, file  = file_exp, quote=F,sep="\t", dec=",",row.names = FALSE,append = TRUE)
```