Advanced statistical tools for SNP arrays : signal calibration, copy number estimation and single array genotyping

Rippe, R.C.A.

Appendices

Fluorescence bias: calibration result tables

A

Affymetrix 100k Hind

Table A.1: Results for linear models fitted on Affymetrix 100k Hind. Global model (3.1) is indicated by G. Local model (3.2) is indicated by L. Improvement of L over G is indicated by D, where D=(G-L)/G *100. Rows: chromosomes. Columns: Global (G), Local (L), Difference (D).

	AA (G)	AB (G)	BB (G)	AA (L)	AB (L)	BB (L)	D(AA)	D(AB)	D(BB)
1	0.064	0.079	0.082	0.056	0.054	0.058	12.636	31.439	29.573
2	0.064	0.080	0.080	0.057	0.055	0.056	11.466	31.410	30.893
3	0.064	0.078	0.082	0.056	0.054	0.055	12.550	31.194	32.255
4	0.064	0.080	0.080	0.056	0.054	0.056	11.266	31.844	30.190
5	0.064	0.079	0.082	0.057	0.055	0.056	11.885	30.018	31.568
6	0.065	0.079	0.086	0.056	0.055	0.057	13.250	30.765	33.516
7	0.066	0.079	0.084	0.058	0.055	0.056	12.369	29.732	33.689
8	0.063	0.077	0.080	0.056	0.055	0.056	11.718	29.350	30.075
9	0.063	0.078	0.082	0.055	0.054	0.054	12.543	30.289	34.396
10	0.064	0.077	0.081	0.056	0.053	0.055	12.484	31.148	32.682
11	0.064	0.079	0.082	0.056	0.054	0.057	12.617	30.849	30.972
12	0.064	0.078	0.081	0.056	0.055	0.055	12.069	30.381	32.807
13	0.064	0.079	0.083	0.056	0.055	0.057	12.168	30.638	32.039
14	0.065	0.080	0.081	0.057	0.055	0.056	12.259	31.339	30.680
15	0.063	0.078	0.081	0.055	0.054	0.056	11.921	30.768	30.721
16	0.064	0.076	0.080	0.056	0.054	0.056	12.611	29.531	29.702
17	0.065	0.077	0.082	0.056	0.055	0.058	13.008	28.968	29.521
18	0.064	0.078	0.079	0.058	0.055	0.057	9.578	29.753	28.458
19	0.065	0.076	0.087	0.057	0.054	0.064	12.138	28.911	26.581
20	0.064	0.079	0.082	0.056	0.055	0.058	13.100	30.296	29.698
21	0.066	0.079	0.086	0.058	0.057	0.063	12.429	28.648	26.434
22	0.070	0.084	0.084	0.060	0.056	0.056	14.437	32.807	33.557

Affymetrix 100k Xba

Table A.2: Results for linear models fitted on Affymetrix 100k Xba. Global model (3.1) is indicated by G. Local model (3.2) is indicated by L. Improvement of L over G is indicated by D, where D=(G-L)/G *100. Rows: chromosomes. Columns: Global (G), Local (L), Difference (D).

	AA (G)	AB (G)	BB (G)	AA (L)	AB (L)	BB (L)	D(AA)	D(AB)	D(BB)
1	0.064	0.079	0.092	0.053	0.050	0.061	17.229	36.306	33.778
2	0.063	0.078	0.091	0.053	0.050	0.060	16.184	36.388	34.178
3	0.062	0.077	0.093	0.050	0.049	0.060	18.109	35.428	35.488
4	0.063	0.076	0.088	0.053	0.050	0.059	14.543	35.152	33.280
5	0.064	0.078	0.093	0.052	0.050	0.060	17.832	35.968	34.820
6	0.065	0.077	0.093	0.054	0.050	0.060	17.791	34.720	35.291
7	0.063	0.079	0.093	0.053	0.050	0.059	16.759	36.039	36.285
8	0.064	0.076	0.092	0.053	0.049	0.061	16.696	34.499	33.859
9	0.063	0.077	0.091	0.052	0.050	0.061	17.069	35.527	33.610
10	0.064	0.077	0.092	0.054	0.050	0.061	16.098	35.668	34.034
11	0.063	0.078	0.090	0.052	0.050	0.060	17.135	35.276	33.261
12	0.062	0.079	0.092	0.051	0.050	0.060	17.405	36.609	34.303
13	0.062	0.075	0.089	0.052	0.049	0.059	15.186	34.321	33.141
14	0.065	0.081	0.092	0.054	0.051	0.060	16.738	37.005	34.370
15	0.063	0.076	0.094	0.053	0.049	0.064	15.142	35.686	31.605
16	0.067	0.078	0.093	0.056	0.049	0.061	17.378	37.282	34.810
17	0.063	0.080	0.093	0.052	0.049	0.061	18.379	38.781	34.504
18	0.063	0.076	0.092	0.052	0.049	0.062	17.531	35.769	32.556
19	0.064	0.080	0.089	0.053	0.054	0.061	18.100	32.901	31.616
20	0.063	0.076	0.092	0.052	0.050	0.061	17.229	33.644	32.981
21	0.063	0.078	0.091	0.054	0.052	0.058	14.833	33.170	36.242
22	0.069	0.083	0.097	0.060	0.055	0.065	12.195	34.227	33.194

Affymetrix 500k NSP

Table A.3: Results for linear models fitted on Affymetrix 500k NSP. Global model (3.1) is indicated by G. Local model (3.2) is indicated by L. Improvement of L over G is indicated by D, where D=(G-L)/G *100. Rows: chromosomes. Columns: Global (G), Local (L), Difference (D).

	AA (G)	AB (G)	BB (G)	AA (L)	AB (L)	BB (L)	D(AA)	D(AB)	D(BB)
1	0.056	0.071	0.085	0.047	0.053	0.066	16.086	26.054	22.280
2	0.056	0.071	0.085	0.047	0.053	0.066	15.852	26.208	22.844
3	0.057	0.070	0.086	0.048	0.052	0.065	15.806	25.745	23.601
4	0.056	0.072	0.085	0.048	0.054	0.067	15.330	24.969	22.022
5	0.056	0.072	0.086	0.047	0.053	0.066	16.106	25.606	23.190
6	0.056	0.071	0.086	0.047	0.053	0.066	16.179	26.166	23.653
7	0.057	0.071	0.086	0.048	0.053	0.066	16.577	25.104	23.919
8	0.056	0.071	0.085	0.047	0.053	0.065	16.588	25.174	23.247
9	0.057	0.071	0.086	0.048	0.053	0.067	16.005	25.574	22.699
10	0.056	0.072	0.085	0.047	0.053	0.066	15.288	26.221	21.970
11	0.057	0.071	0.086	0.048	0.053	0.066	15.990	25.585	22.985
12	0.056	0.071	0.085	0.047	0.053	0.065	15.792	25.442	23.168
13	0.056	0.072	0.085	0.047	0.054	0.065	16.220	25.312	23.620
14	0.057	0.073	0.085	0.048	0.054	0.065	15.966	26.126	23.845
15	0.056	0.071	0.085	0.047	0.053	0.066	16.171	25.817	22.068
16	0.056	0.070	0.084	0.047	0.052	0.064	16.116	26.055	23.642
17	0.056	0.068	0.086	0.046	0.050	0.066	16.258	26.315	23.259
18	0.056	0.071	0.085	0.047	0.053	0.065	15.924	24.556	23.179
19	0.056	0.072	0.085	0.048	0.054	0.066	15.222	24.762	22.596
20	0.056	0.068	0.085	0.048	0.050	0.066	15.138	26.028	21.611
21	0.056	0.072	0.087	0.048	0.054	0.069	15.379	24.683	20.829
22	0.056	0.072	0.086	0.048	0.055	0.068	14.331	24.456	21.003

Affymetrix 500k STY

Table A.4: Results for linear models fitted on Affymetrix 500k STY. Global model (3.1) is indicated by G. Local model (3.2) is indicated by L. Improvement of L over G is indicated by D, where D=(G-L)/G *100. Rows: chromosomes. Columns: Global (G), Local (L), Difference (D).

	AA (G)	AB (G)	BB (G)	AA (L)	AB (L)	BB (L)	D(AA)	D(AB)	D(BB)
1	0.058	0.070	0.081	0.051	0.049	0.062	12.637	29.184	23.774
2	0.059	0.069	0.081	0.051	0.049	0.061	12.732	28.980	24.618
3	0.059	0.068	0.082	0.051	0.049	0.061	13.275	28.572	25.530
4	0.058	0.069	0.081	0.051	0.049	0.061	11.988	28.515	24.337
5	0.059	0.069	0.082	0.051	0.049	0.061	13.166	29.210	25.382
6	0.059	0.069	0.083	0.051	0.049	0.063	13.419	29.090	24.622
7	0.059	0.069	0.082	0.051	0.049	0.061	13.662	28.678	25.219
8	0.059	0.069	0.081	0.051	0.049	0.061	12.746	29.379	24.440
9	0.059	0.069	0.081	0.052	0.049	0.061	12.303	29.300	24.560
10	0.059	0.069	0.080	0.052	0.049	0.061	12.290	29.601	23.334
11	0.059	0.069	0.082	0.051	0.049	0.061	13.327	28.616	24.832
12	0.059	0.069	0.082	0.051	0.049	0.062	12.856	29.549	24.441
13	0.059	0.069	0.080	0.052	0.049	0.060	11.927	29.020	24.631
14	0.058	0.070	0.081	0.051	0.049	0.062	12.346	29.496	23.564
15	0.059	0.068	0.081	0.051	0.049	0.060	12.290	28.291	25.577
16	0.059	0.068	0.081	0.052	0.048	0.061	12.784	29.296	24.526
17	0.059	0.070	0.083	0.052	0.049	0.063	12.979	29.850	23.222
18	0.058	0.070	0.081	0.051	0.050	0.061	12.455	28.635	24.401
19	0.059	0.069	0.083	0.051	0.049	0.064	13.654	29.261	22.808
20	0.059	0.068	0.081	0.051	0.048	0.061	13.047	29.260	24.870
21	0.060	0.070	0.083	0.052	0.050	0.062	13.088	28.143	25.301
22	0.060	0.070	0.080	0.053	0.050	0.061	12.155	28.656	23.503

Affymetrix SNP6.0

Table A.5: Results for linear models fitted on Affymetrix SNP6.0. Global model (3.1) is indicated by G. Local model (3.2) is indicated by L. Improvement of L over G is indicated by D, where D=(G-L)/G *100. Rows: chromosomes. Columns: Global (G), Local (L), Difference (D).

	AA (G)	AB (G)	BB (G)	AA (L)	AB (L)	BB (L)	D(AA)	D(AB)	D(BB)
1	0.063	0.087	0.089	0.052	0.064	0.060	18.148	25.624	32.323
2	0.064	0.087	0.088	0.052	0.066	0.059	18.503	24.126	33.239
3	0.064	0.086	0.089	0.052	0.065	0.059	18.177	24.676	32.900
4	0.065	0.088	0.088	0.053	0.067	0.058	17.883	23.174	33.588
5	0.064	0.086	0.089	0.052	0.065	0.059	17.800	24.526	32.892
6	0.064	0.087	0.089	0.053	0.065	0.060	17.789	24.709	32.729
7	0.064	0.086	0.089	0.052	0.065	0.060	18.187	24.282	32.781
8	0.064	0.086	0.089	0.052	0.064	0.060	18.377	24.935	32.905
9	0.064	0.086	0.089	0.052	0.065	0.060	18.461	24.477	32.663
10	0.063	0.087	0.089	0.051	0.065	0.060	18.111	25.382	32.453
11	0.064	0.088	0.088	0.052	0.067	0.059	18.515	23.982	33.351
12	0.063	0.087	0.088	0.052	0.065	0.060	18.071	24.768	32.577
13	0.064	0.088	0.088	0.053	0.067	0.058	17.412	23.015	33.601
14	0.064	0.088	0.088	0.052	0.066	0.059	18.228	24.975	33.177
15	0.063	0.085	0.089	0.051	0.063	0.060	18.018	25.994	32.369
16	0.062	0.085	0.090	0.051	0.062	0.061	18.160	27.055	32.015
17	0.062	0.086	0.091	0.051	0.064	0.062	18.480	25.964	31.926
18	0.064	0.087	0.088	0.053	0.066	0.059	17.634	24.218	33.173
19	0.063	0.085	0.092	0.050	0.064	0.063	19.627	25.199	31.754
20	0.063	0.087	0.089	0.051	0.064	0.060	18.693	26.814	32.625
21	0.065	0.087	0.088	0.053	0.067	0.059	17.904	22.514	33.437
22	0.064	0.086	0.090	0.051	0.063	0.061	19.160	26.448	31.553

Genotyping: coding scheme

B

Preparation of HapMap data for genotyping comparisons

In this section we describe the data set used in our comparisons, model settings for genotype calling, as well as the translation step to match HapMap calls to our {AA, AB, BB} format.

We compare genotype calls to those of Phase III. We only compare calls to SNPs that have matching 'RSid's. almost half of the total. We disregard the four allelotypes (A,C,G,T) and refer to homozygous genotypes as AA or BB and the heterozygous as AB.

To match our calls to those from HapMap, we need to use the same alphabet. HapMap calls are translated to A and B labels using the following R (R Development Core Team, 2011) code:

```
# create translation vector with default 5
# code contains the SCALA genotype calls
# rssel is a selection vector for matching SNP ids
# from HapMap SNP list, but in the SCALA ordering

# STEP 1:
  d = code[rssel]*0 + 5
# sort scala calls for available rs-ids in HapMap
# rsidt is the working list of HapMap rsids

# STEP 2:
  a = code[rssel][order(rsidt[rssel])]
# get aligned HapMap calls matched to rs-ids.
# hapmap is a dataframe with SNPs in rows,
```

```
# and arrays in colums
# hmsel is the SNP id list for the HapMap ordering

# STEP 3:
  b = hapmap[hmsel,samp+3][order(hapmap$rs[hmsel])]
# now a contains scala calls and
# contains hapmap calls for matching SNP id
# get all heterozygous calls

# STEP 4:
  selhetero = (b!='AA' & b!='CC' & b!='GG' & b!='TT')
# anything not homozygous is translated to 2 (AB)

# STEP 5:
  d[selhetero] = 2
# assign aligned homozygous calls


# STEP 6:
  d[a==1 & !selhetero] = 1
  d[a==3 & !selhetero] = 3
# keep NoCall seperate for later evaluation

# STEP 7:
  d[b=='NN'] = 4
```

Since genotype calls AA from either method are highly unlikely to be mistaken for BB, we can apply the above forced classification from the HapMap homozygous genotype calls into homozygous calls from SCALA.

Waves correction: result tables

C

Fit statistic

Numerical comparison in all following tables are defined as

$$d = \frac{\sum |s_i - z_i|}{n} \tag{C.1}$$

with d the normalized difference between the raw signal s and the smooth profile z (for each SNP i) on a given chromosome.

Output columns

Detailed results are provided for two tumor samples (GBM 139 and GBM 180). Results contain, for each chromosome, the difference for uncorrected data (Raw), after SCALA correction and after NoWaves correction. For both arrays, these tables are given for 4 levels of smoothing: $\lambda \in (1, 10, 100, 1000)$.

C.1 Sample GBM 139

Table C.1: Sample GBM 139; Raw vs SCALA vs NoWaves. All chromosomes for $\lambda = 1$.

Chromosome	Raw 1	SCALA 1	NoWaves 1
1	0.595	0.291	0.291
2	0.595	0.292	0.292
3	0.588	0.279	0.280
4	0.592	0.289	0.290
5	0.596	0.293	0.293
6	0.595	0.288	0.288
7	0.598	0.295	0.296
8	0.586	0.289	0.289
9	0.584	0.290	0.291
10	0.589	0.287	0.287
11	0.591	0.283	0.284
12	0.594	0.290	0.291
13	0.586	0.279	0.280
14	0.583	0.271	0.272
15	0.594	0.286	0.287
16	0.581	0.287	0.287
17	0.581	0.279	0.280
18	0.582	0.281	0.281
19	0.572	0.284	0.285
20	0.591	0.291	0.292
21	0.579	0.279	0.279
22	0.566	0.280	0.280

Table C.2: Sample GBM 139; Raw vs SCALA vs NoWaves.
All chromosomes for $\lambda = 10$.

Chromosome	Raw 10	SCALA 10	NoWaves 10
1	0.598	0.292	0.292
2	0.597	0.293	0.293
3	0.590	0.280	0.281
4	0.594	0.290	0.291
5	0.598	0.294	0.294
6	0.598	0.289	0.290
7	0.601	0.296	0.297
8	0.588	0.290	0.291
9	0.589	0.293	0.294
10	0.592	0.289	0.289
11	0.595	0.285	0.286
12	0.598	0.292	0.293
13	0.589	0.280	0.281
14	0.587	0.273	0.274
15	0.601	0.288	0.289
16	0.587	0.290	0.289
17	0.589	0.282	0.283
18	0.586	0.283	0.284
19	0.583	0.289	0.290
20	0.598	0.295	0.295
21	0.587	0.284	0.284
22	0.583	0.286	0.286

Table C.3: Sample GBM 139; Raw vs SCALA vs NoWaves. All chromosomes for λ = 100.

Chromosome	Raw 100	SCALA 100	NoWaves 100
1	0.600	0.293	0.293
2	0.600	0.293	0.294
3	0.592	0.281	0.282
4	0.596	0.291	0.291
5	0.601	0.294	0.294
6	0.600	0.290	0.291
7	0.603	0.298	0.298
8	0.592	0.292	0.292
9	0.593	0.295	0.297
10	0.594	0.290	0.290
11	0.598	0.287	0.287
12	0.600	0.293	0.294
13	0.592	0.281	0.282
14	0.590	0.274	0.275
15	0.607	0.290	0.291
16	0.591	0.292	0.292
17	0.594	0.285	0.285
18	0.591	0.285	0.286
19	0.592	0.293	0.293
20	0.603	0.297	0.297
21	0.592	0.287	0.287
22	0.594	0.290	0.289

Table C.4: Sample GBM 139; Raw vs SCALA vs NoWaves.
All chromosomes for λ = 1000.

Chromosome	Raw 1000	SCALA 1000	NoWaves 1000
1	0.602	0.293	0.293
2	0.603	0.294	0.295
3	0.594	0.281	0.282
4	0.599	0.291	0.292
5	0.603	0.295	0.295
6	0.602	0.291	0.292
7	0.605	0.298	0.299
8	0.594	0.293	0.293
9	0.598	0.299	0.300
10	0.597	0.291	0.291
11	0.601	0.288	0.289
12	0.602	0.295	0.295
13	0.593	0.282	0.283
14	0.592	0.276	0.277
15	0.610	0.292	0.292
16	0.594	0.293	0.293
17	0.598	0.286	0.287
18	0.594	0.286	0.287
19	0.599	0.295	0.296
20	0.606	0.298	0.298
21	0.596	0.289	0.289
22	0.603	0.293	0.292

C.2 Sample GBM 180

Table C.5: Sample GBM 180; Raw vs SCALA vs NoWaves. All chromosomes for $\lambda = 1$.

Chromosome	Raw 1	SCALA 1	NoWaves 1
1	0.622	0.299	0.300
2	0.620	0.300	0.301
3	0.620	0.297	0.298
4	0.620	0.299	0.300
5	0.624	0.302	0.303
6	0.618	0.296	0.297
7	0.639	0.310	0.311
8	0.611	0.297	0.297
9	0.628	0.312	0.313
10	0.619	0.297	0.298
11	0.615	0.295	0.296
12	0.639	0.315	0.315
13	0.620	0.298	0.299
14	0.620	0.292	0.293
15	0.642	0.312	0.312
16	0.626	0.309	0.310
17	0.611	0.290	0.292
18	0.614	0.294	0.295
19	0.613	0.296	0.297
20	0.620	0.300	0.300
21	0.617	0.301	0.301
22	0.617	0.304	0.304

Table C.6: Sample GBM 180; Raw vs SCALA vs NoWaves.
All chromosomes for $\lambda = 10$.

Chromosome	Raw 10	SCALA 10	NoWaves 10
1	0.624	0.300	0.301
2	0.623	0.302	0.302
3	0.623	0.299	0.300
4	0.622	0.300	0.301
5	0.627	0.304	0.304
6	0.621	0.297	0.298
7	0.642	0.311	0.312
8	0.614	0.298	0.299
9	0.633	0.314	0.315
10	0.622	0.299	0.300
11	0.620	0.297	0.298
12	0.643	0.316	0.317
13	0.624	0.300	0.301
14	0.625	0.294	0.295
15	0.650	0.315	0.315
16	0.631	0.312	0.312
17	0.619	0.294	0.295
18	0.619	0.297	0.297
19	0.626	0.302	0.303
20	0.627	0.303	0.304
21	0.627	0.306	0.306
22	0.634	0.311	0.311

Table C.7: Sample GBM 180; Raw vs SCALA vs NoWaves.
All chromosomes for λ = 100.

Chromosome	Raw 100	SCALA 100	NoWaves 100
1	0.627	0.302	0.302
2	0.625	0.302	0.303
3	0.625	0.300	0.301
4	0.625	0.301	0.302
5	0.629	0.305	0.305
6	0.623	0.298	0.299
7	0.645	0.312	0.313
8	0.617	0.299	0.300
9	0.637	0.316	0.317
10	0.625	0.300	0.301
11	0.623	0.298	0.299
12	0.646	0.318	0.318
13	0.626	0.301	0.302
14	0.628	0.296	0.297
15	0.655	0.317	0.317
16	0.636	0.314	0.314
17	0.624	0.296	0.297
18	0.623	0.299	0.300
19	0.636	0.306	0.307
20	0.632	0.305	0.305
21	0.633	0.310	0.310
22	0.648	0.316	0.317

Table C.8: Sample GBM 180; Raw vs SCALA vs NoWaves.
All chromosomes for λ = 1000.

Chromosome	Raw 1000	SCALA 1000	NoWaves 1000
1	0.629	0.302	0.303
2	0.628	0.303	0.304
3	0.626	0.301	0.301
4	0.626	0.302	0.302
5	0.631	0.305	0.306
6	0.625	0.299	0.300
7	0.647	0.313	0.314
8	0.619	0.300	0.301
9	0.640	0.319	0.319
10	0.628	0.301	0.302
11	0.625	0.299	0.300
12	0.648	0.319	0.320
13	0.628	0.302	0.303
14	0.630	0.297	0.298
15	0.659	0.318	0.319
16	0.639	0.315	0.315
17	0.628	0.298	0.299
18	0.626	0.300	0.301
19	0.644	0.309	0.310
20	0.636	0.306	0.307
21	0.638	0.313	0.314
22	0.658	0.320	0.320

Manual: SCALA Suite

D

D.1 Introduction

This software suite is a collection of programs that were created for and during a PhD project on calibration and genotyping of SNP signals. The whole framework is built on a set of two signals (one for each allele).

Signals from SNP arrays are not perfect; they contain noise. However, in practice this 'noise' has some very structural properties that can be modeled and exploited. It is not hard to imagine that in one SNP array, some SNPs of a particular genotype have a lower signal than other SNPs (of the same genotype). However, we noticed that a SNP with a lower signal behaves similarly in other arrays (of the same platform) as well. Add this to the fact that each array has its own overall signal level and that genotypes are (obviously) expressed in different signal levels for each allele, and there is a strong basis for a model.

The SCALA software models the effects described above. Signals after calibration are much more condensed, which can be beneficial in applications like genotyping (for a single array) and maps of copy numbers and loss of heterozygosity. The latter is not (yet) contained in this suite.

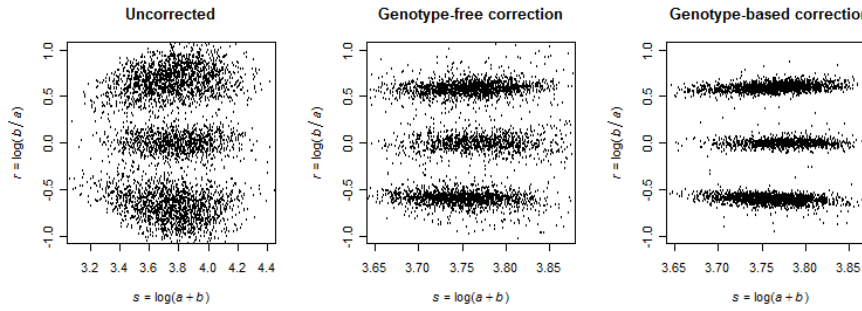
It contains a function for CEL-file conversion to the format used in SCALA (single signal per allele, no probe level information), a function to perform single array genotyping using semi-parametric mixtures on a smoothed 2-dimensional histogram, and a function to obtain signal calibration parameters.

Currently, the software handles mainly Affymetrix CEL files. To be more specific: it handles both enzymes from the 100k platform and both enzymes from the 500k platform, as well as SNP6.0 arrays.

Calibration

To illlustrate the calibration possibilities mentioned above, we provide a graphical example in Figure 1. Starting out with the uncalibrated averaged signals a and b

for allele A and B, we take $s = log(a + b)$ on the horizontal axis and $r = log(b/a)$ on the vertical axis (left panel). This orientation provides three SNP clusters: two for the homozygous genotypes AA (bottom) and BB (top), and one for the heterozygous genotype (middle). Without calibration (after plain signal conversion) this panel shows a lot of noise. However, we can reduce it in the data by using the set of α parameters from the SCALA model (middle panel) or by using Γ from the local model (right panel).



The software can perform global calibration at the conversion stage. The software also provides α sets after model fitting, so that users can perform calibration manually at any later stage.

D.2 The SCALA object class

We defined an object of class SCALA. Not because of object-specific print or plot functions (at this time), but simply to add structure to the results obtained from the different functions contained in this suite. Each of the functions add information to the object. A final object (after conversion, with calibrated signals, and genotyping) has the following structure:

```
> str(scala)

List of 10
$ meta :List of 6
  ..$ fname      : chr "ctr aff 1.CEL"
  ..$ readpath   : chr "D:/Documents/Werk/000 SCALA Suite/01 raw"
  ..$ savepath   : chr "D:/Documents/Werk/000 SCALA Suite/02 arrays"
  ..$ convertDate: chr "2010-12-30 11:21:06"
  ..$ calibrated : logi TRUE
  ..$ callDate   : chr "2010-12-30 12:32:56"
$ chr  : chr [1:262264] "20" "4" "14" "1" ...
$ pos : int [1:262264] 47874178 104894961 51975831 21039991 56554433 ...
$ rsid : chr [1:262264] "rs16994928" "rs233978" "rs2249922" "rs7553394" ...
$ X    : int [1:262264] 267 637 291 2081 772 809 328 421 277 1046 ...
$ Y    : int [1:262264] 1023 776 801 333 989 1043 1398 1359 1183 396 ...
$ Xc   : int [1:262264] 365 722 313 1174 802 804 308 335 303 1157 ...
$ Yc   : int [1:262264] 1234 799 979 315 806 835 1218 1242 1094 364 ...
$ calls: num [1:262264] 3 2 3 1 2 2 3 3 3 1 ...
$ W : num [1:262264, 1:3] 1.96e-10 2.15e-04 2.57e-08 1.00 1.72e-04 ...
 - attr(*, "class")= chr "SCALA"
```

The calibration models and (GUI-based) mapping function currently do not add to the object.

D.3 SCALA.convert: CEL file conversion

Description:

This function converts raw CEL files into aggregated signals X for allele A and Y for allele B.

Usage:

```
SCALA.convert(datatype='Affy250kNSP',calibrate=F,
              readfolder=paste(getwd(),'/01 raw',sep=''),
              savefolder=paste(getwd(),'/02 arrays',sep=''))
```

Arguments:

datatype :	'Affy50kHIND' (default), 'Affy50kXBA'
	'Affy250NSP' , 'Affy250STY'
	'AffySNP6.0'
calibrate :	TRUE (default), FALSE
readfolder :	defaults to getwd()
savefolder :	defaults to getwd()

Details:

The resulting SCALA object is automatically saved to the specified savefolder, to a file that matches [scala$meta$fname].Rdata.

If calibrate is set to T, calibration is indicated and two vectors ($Xc and $Yc) containing the calibrated signals are added after the original signals $X and $Y. The following additions and changes are made:

```
  ..$ calibrated : logi TRUE
..
$ Xc   : int [1:262264] 365 722 313 1174 802 804 308 335 303 1157 ...
$ Yc   : int [1:262264] 1234 799 979 315 806 835 1218 1242 1094 364 ...
```

See also:

SCALA.call, SCALA.global

Examples:

```
scala = SCALA.convert('Affy250kNSP',F,
              readfolder=paste(getwd(),'/01 raw',sep=''),
              savefolder=paste(getwd(),'/02 arrays',sep=''))

str(scala)

List of 7
 $ meta :List of 6
  ..$ fname      : chr "ctr aff 1.CEL"
  ..$ readpath   : chr "D:/Documents/Werk/000 SCALA Suite/01 raw"
  ..$ savepath   : chr "D:/Documents/Werk/000 SCALA Suite/02 arrays"
  ..$ convertDate: chr "2010-12-30 11:21:06"
  ..$ calibrated : logi FALSE
  ..$ callDate   : logi NA
$ chr  : chr [1:262264] "20" "4" "14" "1" ...
$ pos : int [1:262264] 47874178 104894961 51975831 21039991 56554433 ...
$ rsid : chr [1:262264] "rs16994928" "rs233978" "rs2249922" "rs7553394" ...
$ X    : int [1:262264] 267 637 291 2081 772 809 328 421 277 1046 ...
$ Y    : int [1:262264] 1023 776 801 333 989 1043 1398 1359 1183 396 ...
$ calls: logi [1:262264] NA NA NA NA NA NA NA NA ...
 - attr(*, "class")= chr "SCALA"
```

D.4 SCALA.global: calibration

Description:

This function reads all arrays in the readfolder and assumes called genotypes in the SCALA objects.

Usage:

```
params = SCALA.global(filefolder=getwd(), savefolder=getwd(),
                      filename = scala.glob.Rdata, kappa = 1e-8)
```

Arguments:

filefolder :	defaults to getwd()
savefolder :	defaults to getwd()
filename :	defaults to scala.glob.Rdata
kappa :	set value to add to avoid singularity (1e-8)

Details:

The resulting calibration parameters are returned in a separate object, instead of being added to the SCALA object. The reason for this is that the parameters are based on multiple arrays and hence should be added to each array used to obtain the calibration set.

The fields in params match to α, β and γ in the model explained in the appendix. The α values can be used to calibrate the original signal by taking

$$Xc = X/10^{\alpha}.$$

An equivalent approach can be taken for the Y signal. This is the calibration that be performed during CEL file conversion, for the currently implemented platforms.

See also:

SCALA.convert, SCALA.call

Examples:

```
params = SCALA.global()

str(params)

List of 7
 $ celfiles: chr [1:10] "ctr aff 1.CEL.Rdata" "ctr aff 2.CEL.Rdata" ...
 $ alphaX  : num [1:262217] -0.157 -0.0438 -0.0422 0.311 0.0699 ...
 $ alphaY  : num [1:262217] -0.041653 0.016274 -0.062808 0.00015 ...
 $ betaX   : num [1:10] 0.1303 -0.2242 0.1149 0.1248 0.0783 ...
 $ betaY   : num [1:10] 0.114 -0.239 0.1 0.11 0.066 ...
 $ gammaX  : num [1:3] 0.1822 0.0392 -0.2508
 $ gammaY  : num [1:3] -0.2505 0.0922 0.2386
```

D.5 SCALA.call: single array genotyping

Description:

To obtain genotype calls based on a single array, this function 'does the trick'. It uses a mixture of three semi-parametric log-concave densities and classifies each SNP into the cluster with the highest probablility.

Usage:

```
SCALA.call(scala=scala, model='s', plot=F, save=T, xbins = 100,
           ybins = 100, lambda = 10, nit=50, crit=1e-4,
           savefolder=paste(getwd(),'/02 arrays',sep=''))
```

Arguments:

scala :	expects the SCALA object as described above
model :	's': use semi-parametric model, anything other than 's' will revert to a mixture of three parametric regression models using the flexmix package ('s')
plot :	plot single array mixture (FALSE)
save :	save resulting object to file TRUE
xbins :	# of histogram bins to use on x-axis (100)
ybins :	# of histogram bins to use on y-axis (100)
lambda :	sets amount of smoothing in the histogram (10)
nit :	set maximum # of mixture iterations (50)
crit :	sets convergence threshold (1e-4)
savefolder :	defaults to getwd()

Details:

Genotype calls from any source (e.g. HapMap or CRLMM) can be added by simply replacing the $calls vector with the external calls (with AA = 1, AB = 2 and BB = 3).

The result is a change in one meta-tag ($meta$callDate) and addition of two list elements $calls and $W to the SCALA object.

```
..$ callDate   : chr "2010-12-30 12:32:56"
..
$ calls: num [1:262264] 3 2 3 1 2 2 3 3 3 1 ...
$ W : num [1:262264, 1:3] 1.96e-10 2.15e-04 2.57e-08 1.00 1.72e-04 ...
```

See also:

SCALA.convert, SCALA.global

Examples:

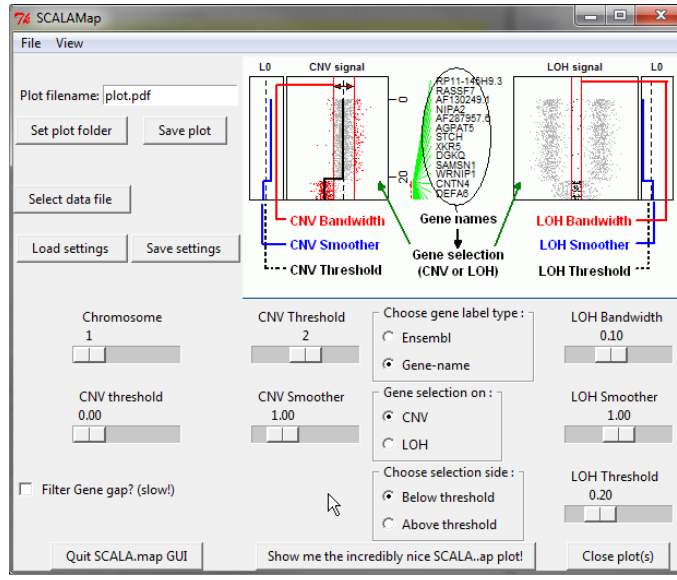
```
scala = SCALA.call(scala, xbins = 75, ybins = 75, lambda = 5)
```

D.6 SCALA.map: CNV / LOH mapping

Description:

The CNV and LOH analyses that are performed detect which genes in either the CNV or LOH signal fall below (or above) the expected threshold number of alleles that is set by the user. The selection results of this detection can be saved either per SNP or per gene. Exported results are saved in .csv format.

The mapping function is fully GUI-controlled (using the rpanel package), not commandline.



Using the GUI the user can

- select the chromosome to analyze,
- choose whether the signal subject to evaluate indicates CNV or lOH,
- where and under what number the analysis figure should be saved,
- choose between Ensembl codes or gene names in selected chromosome regions,
- adapt the signal smoother between power 0 and 2 and
- change plot (title) properties.

Usage:

```
SCALA.map(controls=NA)
```

Arguments:

This function currently only take 1 argument: a saved 'settings' file from a previous analysis.

Details:

The function call simply starts the GUI and doesn't perform any analysis until a SCALA class object is read. If calibrated signals are present, the program uses these automatically, if the $meta$calibrated is set to T.

The exported results file (.csv) contains a number of fields, summarized in the following Table.

See also:

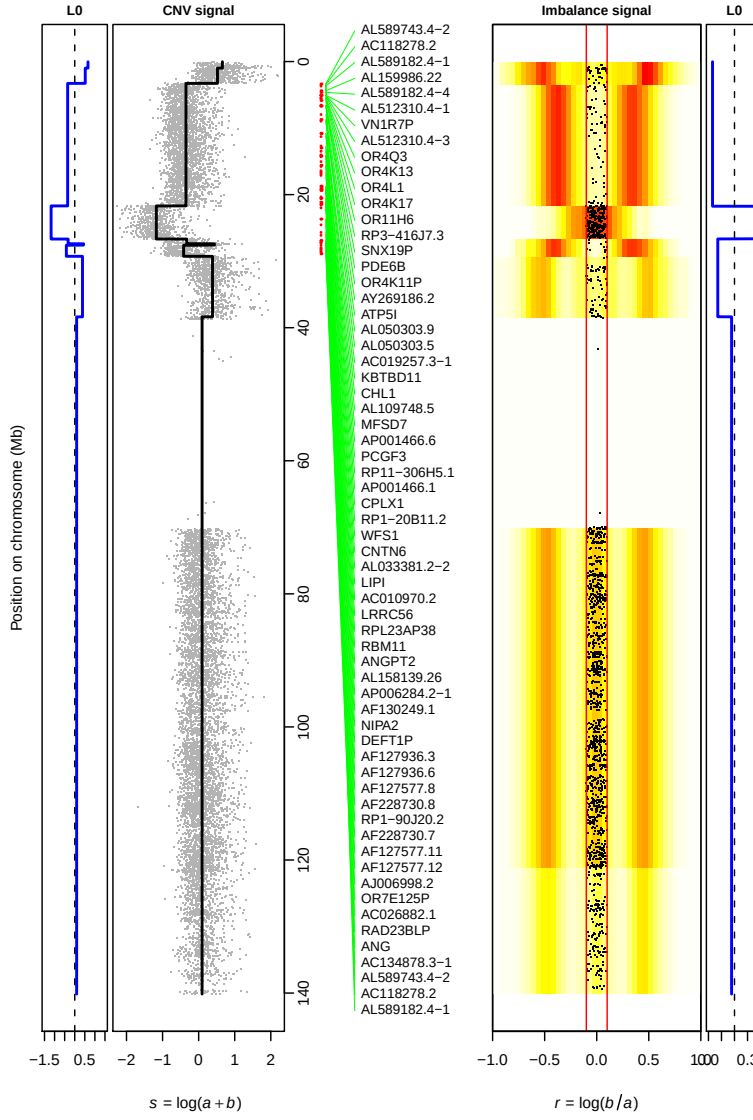
SCALA.convert, SCALA.global, SCALA.call

```
SCALA.map(controls='lastrun.Rdata')
```

The resulting SCALA.map plot:

SNP id :	Database SNP id ('rsid')
CNV sig :	CNV signal value for each SNP
LOH sig :	LOH signal value for each SNP
Position :	SNP position on the chromosome
Chrom :	Chromosome the SNP is located on
Z :	Smoothed CNV value for each SNP
SNP selected :	Indicator whether the SNP exceeds the user-defined threshold
N-level :	Copy Number level for each SNP
GeneBio :	BioMart name of the gene containing the SNP
GeneENS :	Ensembl name of the gene containing the SNP
G-Start :	Starting position of the gene
G-Stop :	Ending position of the gene
Gene selected :	Indicator whether this gene exceeds the user-defined threshold
Mean CNV :	The mean CNV signal in the gene
Mean Z :	Mean CNV smoother value in the gene
Mean LOH :	Mean LOH signal in the gene
Mean G :	Mean LOH smoother value in the gene

GBM 139.CEL chromosome 9



D.7 Appendix: The SCALA model

Theory

The SCALA model aims to find calibration values for the averaged allele intensities for each SNP.

Let $t_{ij} = \log(a_{ij})$, where the logarithms are to base 10. Let the genotypes be coded in the 3-way indicator matrix $H = [h_{ijk}]$, where $k \in \{1, 2, 3\}$ codes for the genotype. $h_{ijk} = 1$ if SNP i on array j has genotype k, otherwise $h_{ijk} = 0$. The first, global, model is written as

$$t_{ij} = \mu + \alpha_i + \beta_j + \sum_{k=1}^{3} \gamma_k h_{ijk} + e_{ij}, \tag{D.1}$$

where μ is the grand mean, α_i the effect of SNP i, and β_j the effect of array j, and γ_k the effect of genotype k. For identifiability, we introduce the constraints $\sum_i \alpha_i = 0$ and $\sum_j \beta_j = 0$. The error $e = [e_{ij}]$ is assumed to have constant variance. The model has one set of genotype parameters (γ) for all SNPs.

A refinement is to have separate genotype parameters for each SNP: $\mathbf{\Gamma} = [\gamma_{ik}]$. We call this the local model, which is specified as

$$t_{ij} = \mu + \beta_j + \sum_{k=1}^{3} \gamma_{ik} h_{ijk} + e_{ij}, \tag{D.2}$$

where we again require that $\sum_j \beta_j = 0$.

Identical models are used for the B allele, with $t_{ij} = \log(b_{ij})$.

Implementation

For the latter model, with appropriate $\boldsymbol{C}$ and $\boldsymbol{D}$, we can write

$$\boldsymbol{t} = \boldsymbol{C}\boldsymbol{\beta} + \boldsymbol{D}\boldsymbol{\gamma} + e \tag{D.3}$$

where $\boldsymbol{\beta}$ contains the n β_j parameters in (D.2) and $\boldsymbol{\gamma}$ = vec($\mathbf{\Gamma}$), i.e. the columns of $\mathbf{\Gamma} = [\gamma_{ik}]$ stacked below each other, and $\boldsymbol{t} = \text{vec}(\boldsymbol{T})$. The structure of $\boldsymbol{C}$ is simple, it can be written as $\boldsymbol{C} = \boldsymbol{I}_n \otimes \mathbf{1}_p$, where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\mathbf{1}_p$ is a vector of ones, of length p. The structure of $\boldsymbol{D}$ is more complex; it consists of n blocks of

diagonal matrices. Each block has three diagonal matrices $\boldsymbol{D_{jk}}$, one for each layer of $\boldsymbol{H}$, and each matrix $\boldsymbol{D_{jk}}$ contains the elements of the jth vector in the kth layer of the 3-way matrix $\boldsymbol{H}$ on its diagonal. Thus, $\boldsymbol{D}$ has dimensions $(n \times p) \times 3p$.

We do not form $\boldsymbol{C}$ and $\boldsymbol{D}$ explicitly. Instead we study the normal equations

$$\begin{bmatrix} \boldsymbol{C'C} & \boldsymbol{C'D} \\ \boldsymbol{D'C} & \boldsymbol{D'D} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C't} \\ \boldsymbol{D't} \end{bmatrix}, \tag{D.4}$$

or

$$\begin{bmatrix} \boldsymbol{V_{11}} & \boldsymbol{V_{12}} \\ \boldsymbol{V_{21}} & \boldsymbol{V_{22}} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \end{bmatrix}, \tag{D.5}$$

where $\boldsymbol{V}_{11} = \boldsymbol{C'C}$, $\boldsymbol{V}_{12} = \boldsymbol{C'D}$, $\boldsymbol{V}_{21} = \boldsymbol{D'C}$, $\boldsymbol{V}_{22} = \boldsymbol{D'D}$, $\boldsymbol{f}_1 = \boldsymbol{C't}$ and $\boldsymbol{f}_2 = \boldsymbol{C't}$. One can prove that $\boldsymbol{C'C} = p\boldsymbol{I_n}$, $\boldsymbol{D'} = \tilde{\boldsymbol{H}}$ and $\boldsymbol{D'D} = \boldsymbol{F}$, where $\tilde{\boldsymbol{H}}$ is a matrix formed by placing the three layers of $\boldsymbol{H}$ below each other. $\boldsymbol{F}$ is a $3p$ by $3p$ diagonal matrix; its first (second, third) p diagonal elements gives, for each SNP, the number of times genotype 1 (2, 3) occurs. Furthermore, $\boldsymbol{C't}$ contains the sums of the columns of $\boldsymbol{T}$, while $\boldsymbol{D't}$ is a stack of three vectors; the first (second, third) vectors contain the sum, per SNP of the elements of $\boldsymbol{t}$ corresponding to genotype 1 (2, 3).

From (D.5) it follows:

$$\hat{\boldsymbol{\gamma}} = \boldsymbol{V}_{22}^{-1}(\boldsymbol{d}_2 - \boldsymbol{V}_{21}\hat{\boldsymbol{\beta}}) \tag{D.6}$$

and hence

$$(\boldsymbol{V}_{11} - \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{V}_{21})\hat{\boldsymbol{\beta}} = \boldsymbol{d}_1 - \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{d}_2. \tag{D.7}$$

Because $\boldsymbol{V}_{22}$ is a diagonal matrix, multiplication by $\boldsymbol{V}_{22}^{-1}$ boils down to dividing the elements of a vector or the rows of a matrix by the corresponding diagonal elements of $\boldsymbol{V}_{22}$. Hence, it is not hard to compute $\boldsymbol{V}_{11} - \boldsymbol{V}_{11}\boldsymbol{V}_{12}^{-1}\boldsymbol{V}_{21}$ and to solve for $\hat{\boldsymbol{\beta}}$, a vector of moderate length. Additional efficiency can be realized by exploiting the way $\boldsymbol{V}_{21}$ is formed. Details on the latter suggestion are considered outside the scope of the current paper.

In this analysis we have ignored the fact that the system in (D.5) is singular, because the condition $\sum_j \beta_j = 0$ is not applied. An easy way out is to demand the minimum-norm solution for $\boldsymbol{\beta}$, by replacing $\boldsymbol{C'C}$ in (D.5) by $\boldsymbol{C'C} + \kappa\boldsymbol{I}$ with κ a small number.

Subject Index

Notes

Curriculum Vitae

Ralph C.A. Rippe was born on February 5, 1982 in Delft. In 2000, he graduated from the Sint Laurens-college (VWO) in Rotterdam. He studied Computer Science at Leiden University, but in 2002, he switched to Psychology. In 2006 he graduated in Methodology and Statistics (*c*um laude). His Master thesis concerned an adaptation of the Multiple Correspondence Analysis algorithm in order to work with datasets containing large design-determined chunks of missing data.

In 2006, after his graduation, he started working as a PhD candidate in the Data Theory Group in the Faculty of Social and Behavioral Science in Leiden. Originally aiming at developing methods for large (wide) datasets from Systems Biology, the project gradually changed its focus to modelling structural properties in SNP signals, after finding many interesting results in a side project. In the course of the project several internal and external cooperations were initiated; among others, with the Department of Neurology at the Erasmus Medical Center in Rotterdam.

During his thesis research, he won several awards. Among these were the Poster Award at the 2nd Channel Network Conference of the International Biometric Society in 2009, the Paper Award at the 24th International Workshop on Statistical Modeling in 2009, and the Presentation Award at the 25th International Workshop on Statistical Modeling in 2010. He was elected as PhD representative of the Interuniversity Research School for Psychometrics and Sociometrics (IOPS) for the period 2008-2010.

Currently, he is a statistician in the department of Clinical Epidemiology in the Leiden University Medical Center.