haplinSlide, GWAS

SAVE, SUMMARIZE AND PLOT RESULTS

Load data:

```
pres.data <- genDataLoad(filename = "data_preprocessed",
 dir.in = "data")
```

- As seen, `haplin` automatically summarizes and plots the results.
- But this can be redone when needed.

Save result:

```
result <- haplin(data = pres.data, markers = 1)
```

- `verbose = F` removes output on EM convergence, etc.
- `printout = F` removes automatic `summary` and `plot`

... not really needed for saving, but looks tidy

SAVE, SUMMARIZE AND PLOT RESULTS

```
result <- haplin(data = pres.data, markers = 1,
  verbose = F, printout = F)
```

Redo summary:

```
summary(result)
```

Redo plot:

```
plot(result)
```

TABULATE RESULT USING haptable

```
haptable(result)
```

```
  marker alleles   counts     HWE.pv Original After.rem.NA
1    rs1     c/G 116/2734 0.8066712      559           475
2   <NA>    <NA>     <NA>         NA      559           475

  After.rem.Mend.inc. After.rem.unused.haplos pv.overall
1                 475                     475  0.8825085
2                 475                     475  0.8825085
```

TABULATE RESULT USING `haptable`

```
  haplos  haplofreq haplofreq.lower haplofreq.upper
1      c 0.03770321      0.02733498      0.05207226
2      G 0.96229679      0.94792774      0.97266502

  reference  RR.est.  RR.lower RR.upper RR.p.value
1         -  1.111558 0.6904018 1.766714     0.6682
2       ref 1.000000 1.0000000 1.000000     1.0000

  RRdd.est. RRdd.lower RRdd.upper RRdd.p.value
1  1.486172   0.188853   12.09931       0.7094
2  1.000000   1.000000    1.00000       1.0000
```

Use `?haptable` to see descriptions of columns

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

MARKER	Name(s) of marker(s) investigated
ALLELES	A listing of the alleles found at each marker
COUNTS	Frequency counts of alleles at each marker
HWE.PV	P-value from Hardy-Weinberg equilibrium test at each marker
ORIGINAL	Number of triads before removal
AFTER.REM.NA	Number of triads after removal of missing
AFTER.REM.MEND.INC.	Number of triads after removal of Mendelian inconsistencies
AFTER.REM.UNUSED.HAPLOS	Number of triads after removal of unused (rare) haplotypes
PV.OVERALL	Overall likelihood ratio p-value (test of all genetic effects combined)

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

HAPLOS	Haplotypes (or single-marker alleles) found during estimation
HAPLOFREQ	Estimated haplotype frequencies
HAPLOFREQ.LOWER	Lower 95% CI for estimated haplotype frequencies
HAPLOFREQ.UPPER	Upper 95% CI for estimated haplotype frequencies
REFERENCE	Reference method. If ref.cat is used, the reference category is labeled ”ref”
RR.EST.	Estimated single dose relative risk
RR.LOWER	Lower 95% CI for single dose relative risk
RR.UPPER	Upper 95% CI for single dose relative risk
RR.P.VALUE	P-values for individual single dose effect
RRDD.EST.	Estimated double dose relative risk
RRDD.LOWER	Lower 95% CI for double dose relative risk
RRDD.UPPER	Upper 95% CI for double dose relative risk
RRDD.P.VALUE	P-values for individual double dose effect

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

maternal = TRUE: **There will be additional columns**

RRM.EST. Estimated single dose relative risk for maternal haplotype

RRM.LOWER Lower 95% CI for single dose relative risk for maternal haplotype

RRM.UPPER Upper 95% CI for single dose relative risk for maternal haplotype

RRM.P.VALUE P-values for individual single dose effect of maternal haplotype

RRMDD.EST. Estimated double dose relative risk for maternal haplotype

RRMDD.LOWER Lower 95% CI for double dose relative risk for maternal haplotype

RRMDD.UPPER Upper 95% CI for double dose relative risk for maternal haplotype

RRMDD.P.VALUE P-values for individual double dose effect of maternal haplotype

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

poo = TRUE:

The RR.est., RR.lower, and RR.upper columns will be replaced by

RRCM.EST. Estimated single dose relative risk, when inherited from the mother

RRCM.LOWER Lower 95% CI for single dose relative risk, when inherited from the mother

RRCM.UPPER Upper 95% CI for single dose relative risk, when inherited from the mother

RRCM.P.VALUE P-values for individual single dose effects, when inherited from the mother

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

poo = TRUE:

The RR.est., RR.lower, and RR.upper columns will be replaced by

RRCF.EST. Estimated single dose relative risk, when inherited from the father

RRCF.LOWER Lower 95% CI for single dose relative risk, when inherited from the father

RRCF.UPPER Upper 95% CI for single dose relative risk, when inherited from the father

RRCF.P.VALUE P-values for individual single dose effects, when inherited from the father

HAPTABLE COLUMNS DESCRIPTIONS (?haptable)

poo = TRUE: **There are the additional columns**

RRCM_RRCF.EST. An estimate of parent-of-origin effect, i.e. the ratio $\mathrm{RRcm/RRcf}$

RRCM_RRCF.LOWER Lower 95% CI for ratio $\mathrm{RRcm/RRcf}$

RRCM_RRCF.UPPER Upper 95% CI for ratio $\mathrm{RRcm/RRcf}$

RRCM_RRCF.P.VALUE P-value for parent-of-origin effect $\mathrm{RRcm/RRcf}$ at that marker

RUNNING haplinSlide

```
result <- haplinSlide(data = pres.data, markers = 1:10,
        table.output = F)
```

- Runs `haplin` on one SNP at a time
- Result is a list of 10 items, each of the 10 full results
- `table.output = F` means that every result is full, not tabulated
- Default is actually `table.output = T`, since `haplinSlide` typically is used for large runs

```
names(result)
```

```
[1] "rs1"  "rs3"  "rs5"  "rs6"  "rs7"  "rs8"  "rs9"  "rs10" "rs11" "
```

Each result can be accessed as usual for a list:

```
result[["rs5"]]
summary(result[["rs5"]])
plot(result[["rs5"]])
```

For instance,

```
summary(result[["rs1"]])
```

```
        ----Child haplotypes----
Haplotype Dose     Relative Risk Lower CI   Upper CI   P-value
c         Single   1.11          0.69       1.77       0.672
c         Double   1.49          0.189      12.1       0.705

G         Single   REF
G         Double   REF
```

RUNNING haplinSlide WITH TABLE OUTPUT

Important: For long runs of `haplinSlide`, default `table.output = T` should be used, like this:

```
result <- haplinSlide(data = pres.data, markers = 1:10)
```

- Runs `haplin` on one SNP at a time
- After each run, `haptable` is automatically applied
- Result is a list of 10 items, each of the 10 haptables

```
names(result)
```

```
 [1] "rs1"  "rs3"  "rs5"  "rs6"  "rs7"  "rs8"  "rs9"  "rs10" "rs11"
```

RUNNING haplinSlide WITH TABLE OUTPUT

```
result[1:2]
```

```
$rs1
  marker alleles   counts    HWE.pv Original After.rem.NA After.rem.Mend.inc. After.rem.u
1    rs1     c/G 116/2734 0.8066712      559          475                 475
2   <NA>    <NA>     <NA>        NA      559          475                 475
  pv.overall haplos  haplofreq haplofreq.lower haplofreq.upper reference  RR.est.  RR.low
1  0.8825085      c 0.03770321      0.02733498      0.05207226         -  1.111558 0.69040
2  0.8825085      G 0.96229679      0.94792774      0.97266502       ref 1.000000 1.00000
  RRdd.est. RRdd.lower RRdd.upper RRdd.p.value
1  1.486172   0.188853   12.09931    0.7054353
2  1.000000   1.000000    1.00000           NA

$rs3
  marker alleles   counts    HWE.pv Original After.rem.NA After.rem.Mend.inc. After.rem.u
1    rs3     A/t 2191/605 0.2340155      559          466                 466
2   <NA>    <NA>     <NA>        NA      559          466                 466
  pv.overall haplos haplofreq haplofreq.lower haplofreq.upper reference   RR.est.  RR.low
1  0.6062371      A 0.7799283       0.7522951       0.8058494       ref 1.0000000 1.00000
2  0.6062371      t 0.2200717       0.1941506       0.2477049         -  0.9127816 0.71202
  RRdd.est. RRdd.lower RRdd.upper RRdd.p.value
1  1.000000  1.0000000   1.000000           NA
2  1.072353  0.6423265   1.794441    0.7805754
```

RUNNING haplinSlide WITH TABLE OUTPUT

- Applying `haptable` to the result produces a single large data frame

	window	row.win	marker	alleles	counts	HWE.pv	Original	After
1	rs1	1	rs1	c/G	116/2734	0.8066712	559	
2	rs1	2	<NA>	<NA>	<NA>	NA	559	
3	rs3	1	rs3	A/t	2191/605	0.2340155	559	
4	rs3	2	<NA>	<NA>	<NA>	NA	559	
5	rs5	1	rs5	a/T	1056/1770	0.2701807	559	
6	rs5	2	<NA>	<NA>	<NA>	NA	559	
7	rs6	1	rs6	c/G	252/2592	0.2987255	559	
8	rs6	2	<NA>	<NA>	<NA>	NA	559	
9	rs7	1	rs7	c/G	1127/1693	0.5638047	559	
10	rs7	2	<NA>	<NA>	<NA>	NA	559	
11	rs8	1	rs8	A/g	2051/751	0.6198633	559	
12	rs8	2	<NA>	<NA>	<NA>	NA	559	

Recommendations:

- Use the default `table.output = T`
- Use the (non-default) `response = "mult"`, i.e. a multiplicative response model
- ... and set `reference = "ref.cat"`
  (which is standard when `response = "mult"`)
- Use multi-core runs using, e.g. `cpus = 4`

Example:

```
result <- haplinSlide(data = pres.data, markers = 1:200,
 response = "mult", reference = "ref.cat", cpus = 4)
```

```
result[["rs100"]]
```

```
result <- haptable(result)
result[result$window == "rs100", c("window", "row.win",
 "marker", "alleles", "RR.est.", "RRdd.est.", "RR.p.value")]
```

```
    window row.win marker alleles RR.est. RRdd.est. RR.p.value
199   rs100       1   rs100     a/T 1.01052  1.021151  0.9096187
200   rs100       2   <NA>    <NA> 1.00000  1.000000         NA
```

The data frame `result` can then be saved as for any other data frame, using e.g. `write.table`.

haplinSlide AND GWAS

- Haplin was built for “Finesse”, not for speed :-)
- Patience is needed
- Do first a check run on, say, 1000 SNPs
- Preferably split into e.g. one chromosome at a time by
  - using the `markers` argument
  - ...or extract from file in advance using e.g. the `genDataGetPart` function
- Remember the argument settings:

  `table.output = T` (default), `response = "mult"`,

  `reference = "ref.cat"`, `cpus = 4` (or something much larger...)
- Perfectly doable on a desktop PC
- ... but may take a few days for a full GWAS
- The Haplin web page has instructions for running on large clusters