# SNPAssayR: four SNP assay workflows

SNPAssayR authors

## 1 Scope and input

The package runs offline using base R for core analysis. Shiny and bslib are optional interface dependencies. The main tutorial uses a 550-base synthetic locus and actual 22-base oligos. Short artificial coordinate fixtures follow separately. Neither example type represents a naturally observed locus or a validated marker. Use biologically justified constraints and independent laboratory validation for real designs. No KASP workflow is included.

```
> library(SNPAssayR)
> old_options <- options(width = 68)
```

Sequences may be canonical/IUPAC strings or FASTA text. File import in the API uses `paste(readLines(path), collapse="
n")`. Multiple FASTA records require an exact first-token identifier; records are never concatenated. Only case/whitespace normalization is performed and recorded. SNP coordinates are one-based and alleles must be distinct A/C/G/T, matching the reference base. Supplied copies must be named; they are unaligned comparison sequences.

```
> l <- read_locus(
+   ">target\nACGTTGAATTCAAAAAAGTC\n>copy\nACGTTGAATTCAAAAAAGTC",
+   "fixture", record = "target")
> s <- validate_snp(l, 6, "G", "A")
> stopifnot(s$alternative_sequence == "ACGTTAAATTCAAAAAAGTC")
```

## 2 A complete 550-base teaching workflow

The same synthetic locus and C275G SNP demonstrate all four methods. The locus was constructed for teaching, has no biological accession and has no experimental validation. It is supplied as a readable FASTA file; no download is needed. All primers in this walkthrough are 22 nucleotides long, within the 20–22 nucleotide range. Normal GC/Tm filters and the five-degree primer-Tm spread are retained. To search all three lengths, use `primer_lengths=20:22` and inspect the resulting search limits. The following script runs the four public design functions and checks the expected arithmetic:

```
> source(system.file("examples", "tutorial_550bp.R",
+                    package = "SNPAssayR"))
```

The executable script is installed with the package. Its design calls are:

```
# tutorial_snp: the 550 bp FASTA, position 275, C -> G.
e <- restriction_enzymes()
caps <- design_caps(tutorial_snp,
  e[e$name == "MfeI-HF", , drop = FALSE],
  primer_lengths=22, product_min=120, product_max=300,
  max_candidates=1)
dcaps <- design_dcaps(tutorial_snp,
  e[e$name == "EcoRI", , drop = FALSE],
```

```
  primer_lengths=22, primer_gaps=0,
  product_min=120, product_max=300, max_candidates=1)
arms <- design_arms(tutorial_snp, strands="+",
  primer_lengths=22, product_min=120, product_max=300,
  max_candidates=1)
tetra <- design_tetra_arms(tutorial_snp,
  primer_lengths=22, product_min=100, product_max=350,
  min_size_difference=20, pool_size=35,
  max_combinations=20000, max_candidates=1)
```

The local sequence at positions 270–275 is CAATTC in the reference and CAATTG in the alternative. MfeI-HF therefore cuts the alternative product in CAPS. The dCAPS forward primer changes C270 to G, creating GAATTC in the reference product and GAATTG in the alternative. This intentional mismatch is five bases from the primer's 3-prime end; the SNP itself is preserved.

```
> tutorial_primers[, c("method", "role", "sequence_5prime_to_3prime")]

   method                role sequence_5prime_to_3prime
1    caps              forward    GCGTACGATGCTACGTCCAATT
2    caps              reverse    TGGTCATTACAATCGGGCGGCG
3   dcaps              forward    GCGTACGATGCTACGTCGAATT
4   dcaps              reverse    TGGTCATTACAATCGGGCGGCG
5    arms   reference_specific    CGTACGATGCTACGTCCAATTC
6    arms alternative_specific    CGTACGATGCTACGTCCAATTG
7    arms               common    ATGGTCATTACAATCGGGCGGC
8   tetra        outer_forward    ATAGTGGCGTGTATCTGTCGCC
9   tetra        outer_reverse    GCGCAACTTTCAAAAGGATGAT
10  tetra        inner_forward    CGTACGATGCTACGTCCAATTC
11  tetra        inner_reverse    AAACCCAGACTACACAACGTTC

> range(tutorial_primers$tm_celsius)

[1] 55.50377 62.75380

> tutorial_assays$tetra$candidates[[1]]$intended_products

      product start end length_bp
1     control   174 353       180
2   reference   254 353       100
3 alternative   174 296       123
```

CAPS: reference 120 bp uncut; alternative 18 + 102 bp. dCAPS: reference 18 + 102 bp; alternative 120 bp uncut. Conventional ARMS has separate 120 bp allele reactions. Tetra-primer ARMS has an outer control of 180 bp, a reference product of 100 bp and an alternative product of 123 bp, conditional on successful amplification and discrimination. The 18 bp digest fragment may be difficult to observe experimentally; neither its detection nor separation of 102 and 120 bp bands is established by these calculations. These examples teach product reconstruction, not a validated gel.

Return limits are deliberately small to keep the tutorial bounded. The first candidate is an enumeration result, not a globally optimal assay. Search truncation is retained and must be inspected. For actual loci, search a suitable panel and inspect multiple candidates, supplied-copy screens and quality metrics. No genome-wide specificity evidence is supplied for this artificial locus.

```
tutorial_assays$caps$search
rank_assays(tutorial_assays$caps)$components
assay_tables(tutorial_assays$dcaps)$primers
tutorial_assays$dcaps$candidates[[1]]$intended_product$edits
```

```
tutorial_assays$arms$candidates[[1]]$ideal_reaction_profiles
tutorial_assays$tetra$candidates[[1]]$ideal_genotype_profiles
draw_scene(assay_scene(tutorial_assays$tetra))
draw_scene(band_scene(tutorial_assays$caps))
```

In Shiny, select a "550 bp" teaching example, validate the input and analyze. The same sequence and settings are used. Replace the synthetic locus with your own sequence and confirm the reference allele at the chosen coordinate.

# 3 Short coordinate fixtures

The examples below isolate geometry with deliberately short primers. They are unit-test illustrations and should not be used as laboratory primer designs.

# 4 CAPS

Natural recognition differences are screened on complete intended allele products. All internal supported cuts are included. Twenty curated enzyme records are supplied; only palindromic recognition with supported symmetric internal strand cuts is accepted. Fragment lengths sum to the linear product length. This is theoretical complete digestion, not digestion efficiency.

```
> caps <- design_caps(s, restriction_enzymes()[1, , drop = FALSE],
+   primer_lengths = 3, product_min = 20, product_max = 20,
+   gc_range = c(0, 1), tm_range = c(-250, 100),
+   max_tm_difference = 100, screening_mismatches = 0)
> stopifnot(length(caps$candidates) == 1)
> caps$summary

  candidate_id enzyme forward_primer reverse_primer start end
1   CAPS000001  EcoRI            ACG            GAC     1  20
  length_bp tm_difference best_exclusive_band_distance_bp
1        20      6.955997                              14

> caps$candidates[[1]]$digests$reference$fragments

  fragment_id start end length_bp
1           1     1   6         6
2           2     7  20        14
```

# 5 dCAPS

One intentional primer substitution creates a SNP-spanning diagnostic site. Mature products are reconstructed before full digestion. Designs that overwrite the SNP or have unsupported overlapping primers are rejected. Mismatched-template thermodynamics and extension are not modeled.

```
> d <- validate_snp(read_locus("ACGGACTTCAAAGTC", "fixture"), 9, "C", "G")
> dcaps <- design_dcaps(d, restriction_enzymes()[1, , drop = FALSE],
+   primer_lengths = c(3, 8), primer_gaps = 0,
+   product_min = 15, product_max = 15, gc_range = c(0, 1),
+   tm_range = c(-250, 100), max_tm_difference = 100)
> stopifnot(length(dcaps$candidates) == 1)
> dcaps$candidates[[1]]$intended_product$edits
```

```
   primer strand template_coordinate product_coordinate
1 forward      +                   6                  6
  primer_position_5prime primer_position_3prime template_base_plus
1                      6                      3                 C
  product_base_plus old_primer_base new_primer_base
1                 A               C               A
```

# 6 Conventional ARMS-PCR

Two allele-specific primers share a common primer in separate reactions. Terminal SNP matches and optional intentional mismatch positions are explicit. Conditional profiles assume ideal allele discrimination. Absence of a band alone cannot establish genotype; no independent amplification-control pair is designed.

```
> a <- validate_snp(read_locus("ACGAAAGTC", "fixture"), 3, "G", "T")
> arms <- design_arms(a, strands = "+", primer_lengths = 3,
+   product_min = 9, product_max = 9, gc_range = c(0, 1),
+   tm_range = c(-250, 100), max_tm_difference = 100)
> stopifnot(length(arms$candidates) == 1)
> arms$candidates[[1]]$ideal_reaction_profiles[,
+   c("genotype", "reaction", "ideal_product_present", "ideal_sizes_bp")]
```

```
  genotype    reaction ideal_product_present ideal_sizes_bp
1       AA   reference                  TRUE             9
2       AA alternative                 FALSE
3       AB   reference                  TRUE             9
4       AB alternative                  TRUE             9
5       BB   reference                 FALSE
6       BB alternative                  TRUE             9
```

# 7 Tetra-primer ARMS-PCR

Two outer and two opposing allele-specific inner primers produce a conditional outer control and two diagnostic sizes. All six pairwise compatibility screens are retained. Product bounds include the outer control and both diagnostic products. Risk screens cover 1–10,000 bp independently of these design bounds. Arithmetic size separation is not an empirical gel-resolution model.

```
> t <- validate_snp(read_locus("ACGTCAGTACGATCGTACGTTAGCGATCGA",
+                              "fixture"), 14, "C", "A")
> tetra <- design_tetra_arms(t, primer_lengths = 3,
+   product_min = 12, product_max = 25, gc_range = c(0, 1),
+   tm_range = c(-250, 150), max_tm_difference = 200,
+   min_size_difference = 1, pool_size = 2, max_candidates = 1)
> stopifnot(length(tetra$candidates) == 1)
> tetra$candidates[[1]]$intended_products
```

```
      product start end length_bp
1     control     4  23        20
2   reference    12  23        12
3 alternative     4  16        13
```

# 8 Inspection and reproducibility

Rankings are deterministic within one result and express preferences, not success probabilities. Inspect search limits, evidence and supplied-copy warnings. A potential product is sequence-compatible within the bound, not confirmed PCR. Genome-wide specificity is unassessed. Length graphics predict neither band intensity nor detection. Perfect-match primer Tm uses the documented nearest- neighbor model; interaction scores are transparent sequence heuristics.

```
> rank_assays(caps)$components[,
+   c("candidate_id", "quality", "evidence", "score", "rank")]
```

```
  candidate_id  quality evidence    score rank
1   CAPS000001 7.956482        4 15.06621    1
```

```
> bundle <- assay_bundle(caps, selected = 1)
> destination <- tempfile("snpassay-export-")
> export_assay(bundle, destination, figure_formats = "pdf")
> rerun <- replay_assay(readRDS(
+   file.path(destination, "analysis-bundle.rds")))
> stopifnot(identical(rerun, caps))
> unlink(destination, recursive = TRUE)
> options(old_options)
```

For interactive analysis, run `SNPAssayR::run_app()` with optional dependencies installed. Import, validate, configure, Analyze, select a candidate, inspect linked plots and export. Changing analysis inputs invalidates old results and downloads even if the former setting is restored; analyze again.

Computational agreement against finite independent references and regression fixtures is distinct from experimental validation. No wet-lab validation has been performed or verified. Installed methods under `system.file("methods", package="SNPAssayR")` specify coordinates, chemistry scope, search caps and uncertainty. Function help gives all parameters.

# 9 Authorship and copyright

Prashant Yadav, Vinod Kumar and Vijay Veer Singh are the package authors. Prashant Yadav is the designated maintainer. The recorded copyright holder is ICAR-Indian Institute of Rapeseed-Mustard Research (ICAR-IIRMR). The package is distributed under GPL-3. Authorship credit is retained in the software citation.

# 10 How to cite SNPAssayR

Credit the software using the version-specific citation returned by:

```
> citation("SNPAssayR")

To cite SNPAssayR in publications, use:

  Yadav P, Kumar V, Singh VV (2026). _SNPAssayR: SNP Genotyping Assay
  Design and Visualization_. R package version 1.0.1.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {SNPAssayR: SNP Genotyping Assay Design and Visualization},
    author = {Prashant Yadav and Vinod Kumar and Vijay Veer Singh},
    year = {2026},
    note = {R package version 1.0.1},
  }
```

The citation is a software manual entry for the three package authors, year 2026 and version 1.0.1. No published package article or DOI is claimed. Cite the method papers relevant to your analysis separately; they do not replace the software citation. A BibTeX record can be obtained with `toBibtex(citation("SNPAssayR"))`.