

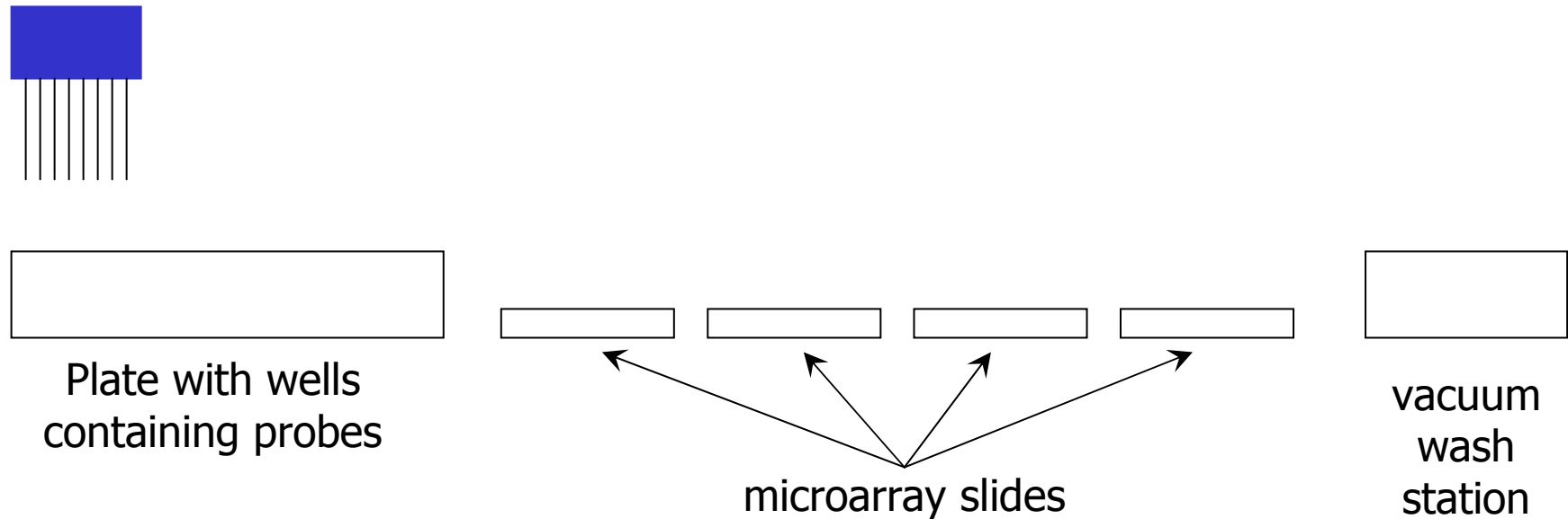
# R: Microarray data structures

Perry Moerland

Bioinformatics Lab, KEBB, AMC

`p.d.moerland@amc.uva.nl`

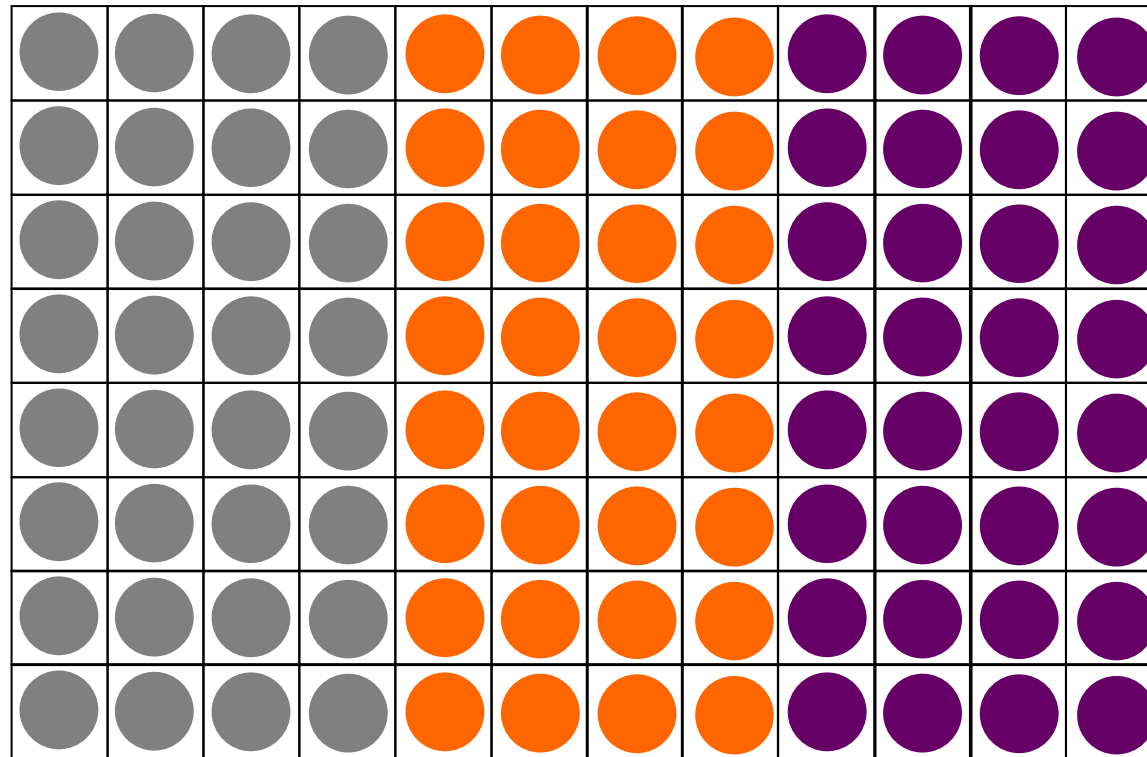
# cDNA microarray: printing process (side view)



# cDNA microarray: spotting probes

## 8 X 4 print head

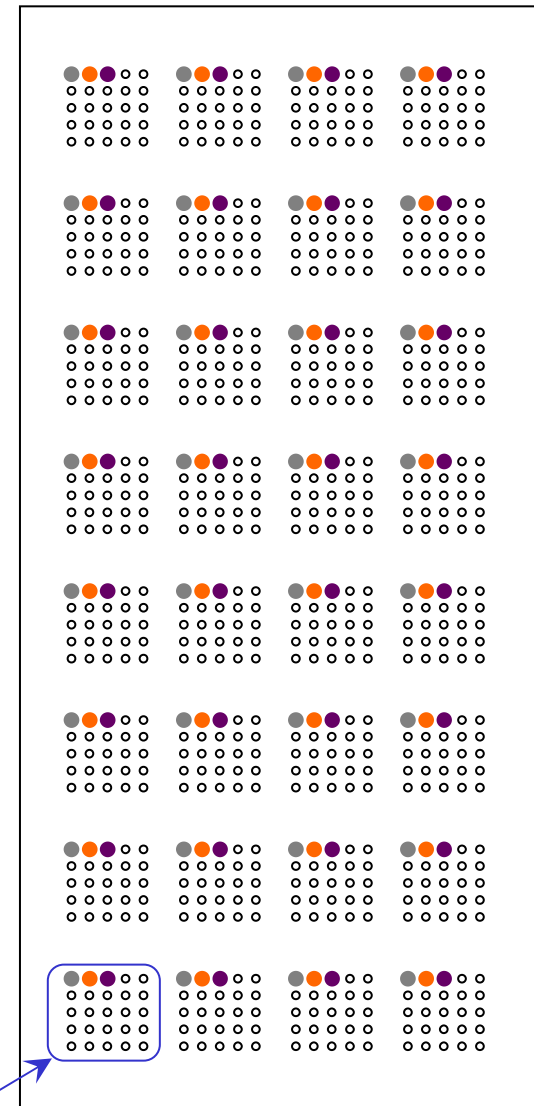
plate with wells holding probes in solution



All spots of the same color are made at the same time

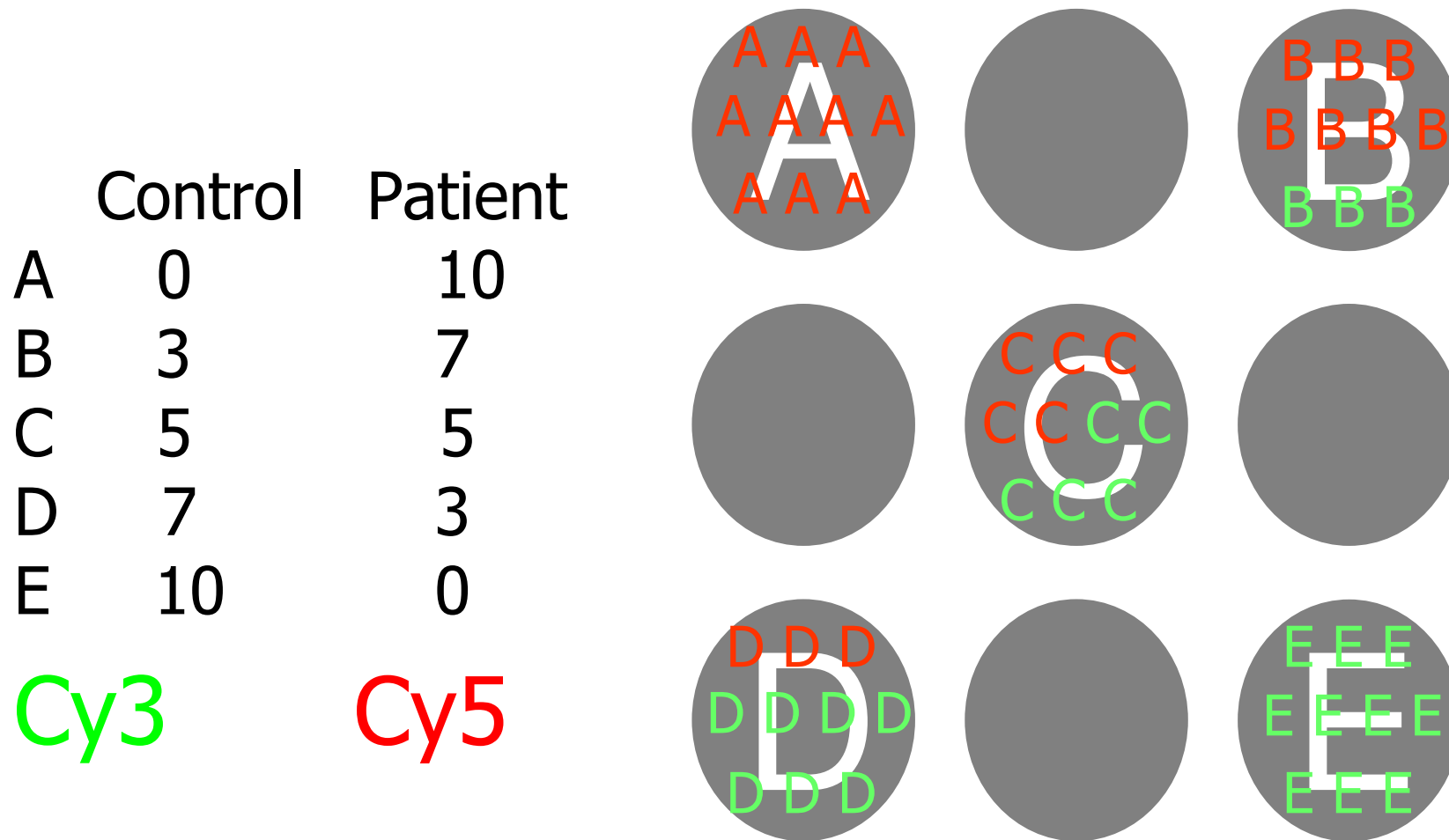
All spots in the same sector are made by the same pin

microarray slide



# Array hybridization

Base pairing between probe & target



## `marray` package

The `marray` package uses four basic classes for data from a 2-dye microarray experiment

- **`marrayLayout`**: describes the geometry of the array
- **`marrayInfo`**: holds sample information or gene information
- **`marrayRaw`**: holds the raw array data
- **`marrayNorm`**: holds array data after normalization

# marrayLayout

marray uses an `marrayLayout` object to describe the geometry using five numbers:

- **maNgr** : number of grid rows
- **maNgc** : number of grid columns
- **maNsr** : number of spot rows
- **maNsc** : number of spot columns
- **maNspots** : number of spots

} slots

$$\text{maNspots} = \text{maNgr} \times \text{maNgc} \times \text{maNsr} \times \text{maNsc}$$

Access with @: `layout@maNspots`

## `marrayLayout`: additional slots

A `marrayLayout` object may also include three additional vectors

- **maSub** : a logical vector - are we currently interested in this spot?
- **maPlate** : which plate did the robot get this spot from?
- **maControls** : what kind of material is spotted here?

## `marrayLayout` methods

Methods to compute the following quantities that are not stored in the object:

- **maPrintTip** : vector of print tips for the spots
- **maGridCol** : vector of grid column locations
- **maGridRow** : vector of grid row locations
- **maSpotCol** : vector of spot column locations
- **maSpotRow** : vector of spot row locations

# marrayRaw slots

Raw expression data from 2-dye microarrays is stored as an `marrayRaw` object, which contains:

- Four matrices of raw data (`maRf`, `maGf`, `maRb`, `maGb`) with red (R) and green (G) foreground (f) and background (b) estimates.
- An optional matrix (`maW`) of spot quality weights.
- `maLayout` containing the array layout (class: `marrayLayout`)
- `maGnames` containing the gene information (class: `marrayInfo`)
- `maTargets` containing the sample information (class: `marrayInfo`)



## `marrayRaw` methods

- **maA** : matrix of log intensities
- **maM** : matrix of log ratios
- **maLR** : matrix of background-corrected red log intensities
- **maLG** : matrix of background-corrected green log intensities

## marrayNorm slots

Processed expression data from 2-dye microarrays is stored as an `marrayNorm` object. These contain copies of the `maW`, `maLayout`, `maGnames`, and `maTargets` objects from the raw source data. In place of the raw measurements, these objects contain

- **maA** : matrix of average log intensities
- **maM** : matrix of log ratios
- **maMloc** : localization normalization values
- **maMscale** : scale normalization values

# limma data structures

The `limma` package in BioConductor provides a different set of tools for 2-dye microarrays.

- **RGList** : raw microarray data as a list of arrays containing
  - Four matrices, `R`, `G`, `Rb`, `Gb`, containing measurements.
  - Optional components `weights`, `printer`, `genes`, `targets`.
- **MAList** : processed microarray data as a similar list with `M` and `A` components

# Conversion: `convert` package

Conversion via `as()`

- `RGList` (`limma`)
- `MAList` (`limma`).
- `marrayRaw` (`marray`)
- `marrayNorm` (`marray`).
- `ExpressionSet` (`Biobase`)

The `convert` package provides conversion to and from `RGList` **and** `marrayRaw`, to and from `MAList` **and** `marrayNorm`, **and from** `MAList` **and** `marrayNorm` **to** `ExpressionSet`