

Slice & Dice

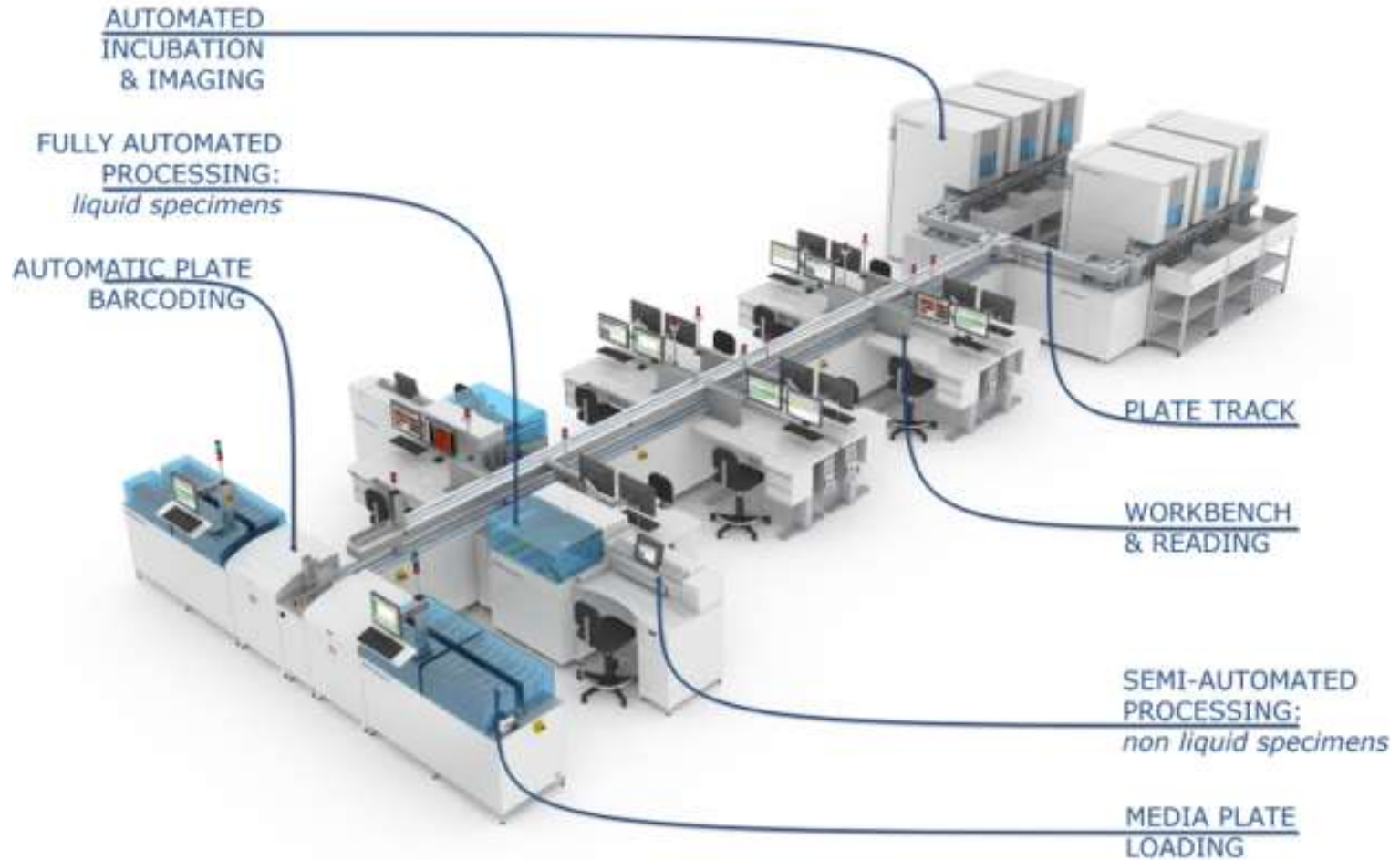
Cell line imaging with 100% PostgreSQL

Dr. Ernst-Georg Schmid
pgconf.eu Vienna 2015

Old Lab: Lots of Bottles



New Lab: Lots of Robots



Where the cells grow...



File formats

Coordinates file:

Unimportant header line

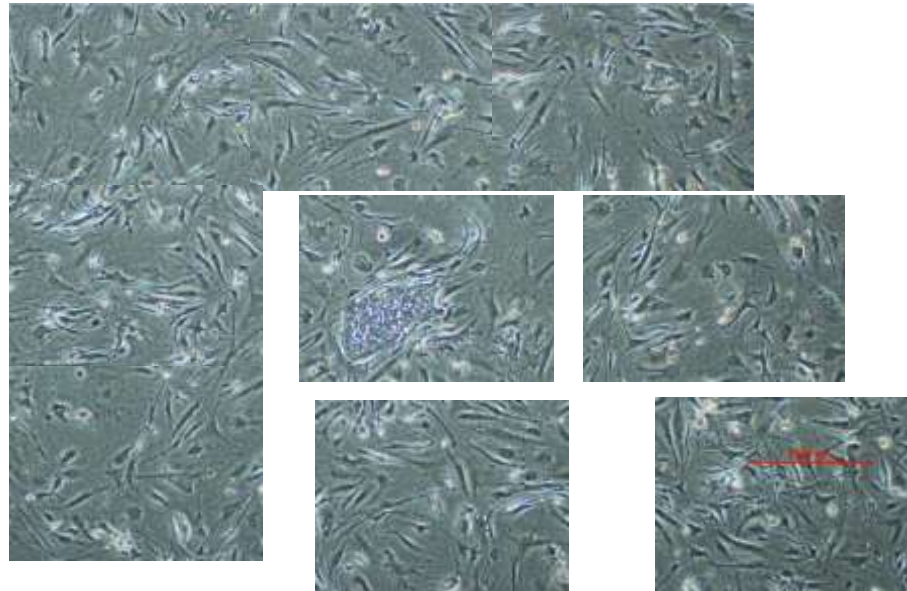
```
1, 8, 512, 512 ←  
\1-1\0\tile-0-0, 0, 33528  
\1-1\0\tile-0-1, 33528, 41438  
\1-1\0\tile-1-0, 74966, 40772  
\1-1\0\tile-1-1, 115738, 54786  
\1-1\0\tile-0-6, 170524, 52274  
\1-1\1\tile-0-0, 222798, 46925  
\1-1\0\tile-0-7, 269723, 33430  
\1-1\0\tile-1-6, 303153, 77888  
\1-1\0\tile-0-2, 381041, 59736  
\1-1\0\tile-1-7, 440777, 48756  
\1-1\0\tile-0-3, 489533, 77646  
\1-1\1\tile-0-3, 567179, 57215  
...
```

+

Data file:
Binary blob
containing all
images as tiles

Objective

Load imagesets into PostgreSQL and extract individual tiles by coordinates -> reconstruct the whole image. 100% SQL due to analysis tool limitations



Imagesets table

```
CREATE TABLE imagesets (  
    barcode INTEGER, --Plate ID  
    tileset_id OID, --LOB ID  
CONSTRAINT pk_imagesets UNIQUE (barcode) );
```

Tiles table

```
CREATE TABLE tiles (  
  path TEXT, --Coordinates  
  "offset" BIGINT, --Start in Bytes  
  length INTEGER, --Size in Bytes  
  barcode BIGINT, --Plate ID  
  CONSTRAINT tiles_fk FOREIGN KEY (barcode)  
    REFERENCES everything.imagesets (barcode)  
    ON DELETE CASCADE  
    ON UPDATE RESTRICT  
  NOT DEFERRABLE);
```

Tiles view

```
CREATE VIEW v_tiles (  
    imageset_id,  
    "row",  
    col,  
    type,  
    pos1,  
    pos2,  
    tiledata)  
AS  
SELECT t.imageset_id,  
    split_part(split_part(t.path, '\'::text, 2), '-'::text, 1)::integer AS "row",  
    split_part(split_part(t.path, '\'::text, 2), '-'::text, 2)::integer AS col,  
    split_part(t.path, '\'::text, 3)::integer AS type,  
    split_part(split_part(t.path, '\'::text, 4), '-'::text, 2)::integer AS pos1,  
    split_part(split_part(t.path, '\'::text, 4), '-'::text, 3)::integer AS pos2,  
    lo_get(i.tileset_id, t."offset", t.length) AS tiledata  
FROM tiles t  
    JOIN imagesets i ON t.imageset_id = i.id;
```

Loading data

```
lo_import (<images_file>);  
COPY ... FROM  
<coordinates_file>  
(HEADER); --skip first row
```

Look Ma, fully transactional!

And no single line of external code!

Limitations

- Cannot delete individual tiles, only tilesets
- Max. 32 TB per database

Slice & Dice

Cell line imaging with 100% PostgreSQL

Dr. Ernst-Georg Schmid
pgconf.eu Vienna 2015

References

- https://commons.wikimedia.org/wiki/File:The_Science_Laboratory.jpg
- <http://www.bd.com/scripts/europe/labautomation/productsdrilldown.asp?CatID=455&SubID=1836&siteID=20309&d=&s=europe%2Flabautomation&sTitle=Lab+Automation&metaTitle=Total+Lab+Automation&dc=europe&dcTitle=Europe>
- https://commons.wikimedia.org/wiki/File:Petri_dish_at_the_Pacific_Northwest_National_Laboratory.jpg
- https://commons.wikimedia.org/wiki/File:96_well_plastic_cell_culture_plate-03.jpg
- http://www.stembook.org/sites/default/files/protocols/Borowski_F02.jpg