DEMETRIX

F# for Biotech — Stranger Things edition

Darren Platt @dplattsf

20201021

Copyright © 2020 Demetrix, Inc. All rights reserved

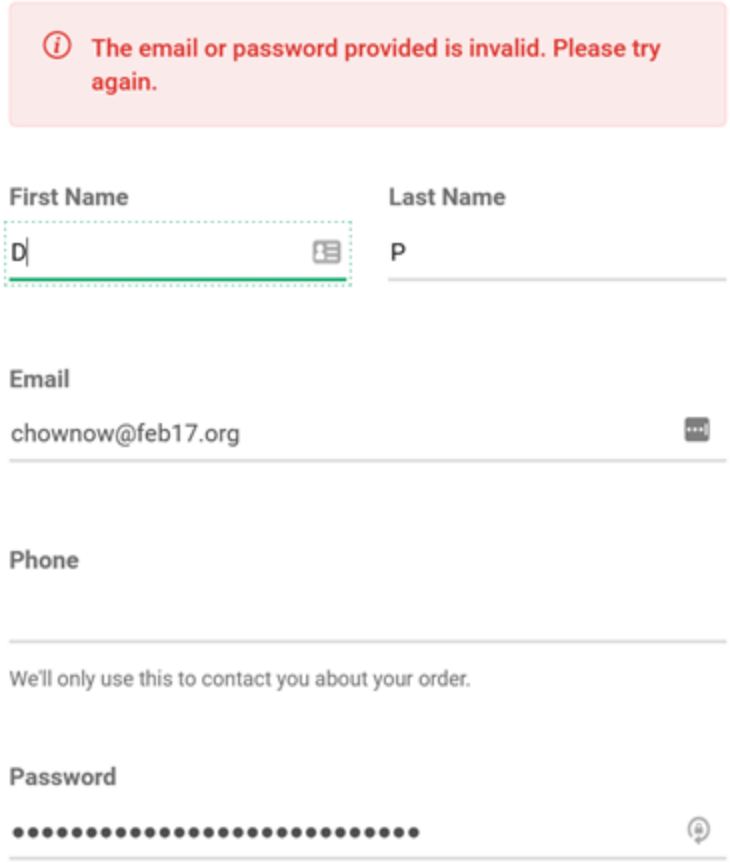
Thanks!

- Developers
  - Garrett Birkel
  - Dmitry Morozov
  - Gien Verschatse
  - Max Grebeniuk
  - Marcell Legeza
  - Alfonso Garcia
  - Olya Samusik
  - Finn Van Krieken
  - Dave Thomas
  - Nick Pinto
- Community
  - Alfonso Garcia [Fable and solid advice]
  - Don Syme [Language + Type provider support]
  - Postgres Npgsql maintainers [improvements in type safety]
  - Dustin Moris Gorski [Giraffe]
  - Elliot Menschik, Paul Underwood [AWS Biotech support]
  - Krzysztof-Cieslak [Ionide]
  - Matthias Brandewinder [Recruiting]
  - Chris Macklin [Genotype Specification Language collaboration]
  - A ton of people I have missed..

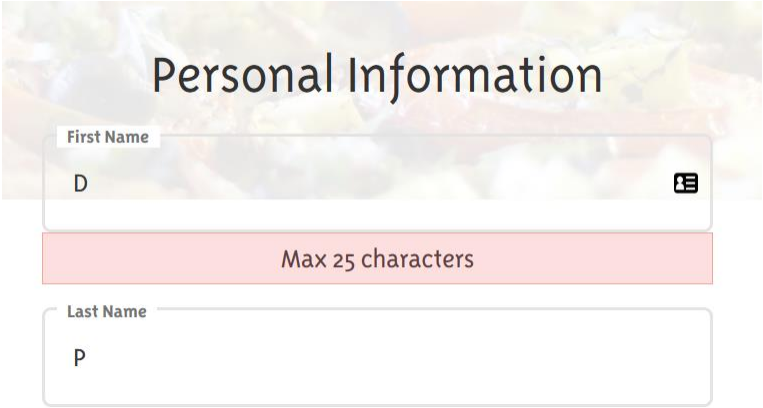
Copyright © 2020 Demetrix, Inc. All rights reserved.

Delivering correct code is difficult (food UI metric)

Burrito ordering new customer sign up form
How can a new password or email be invalid?

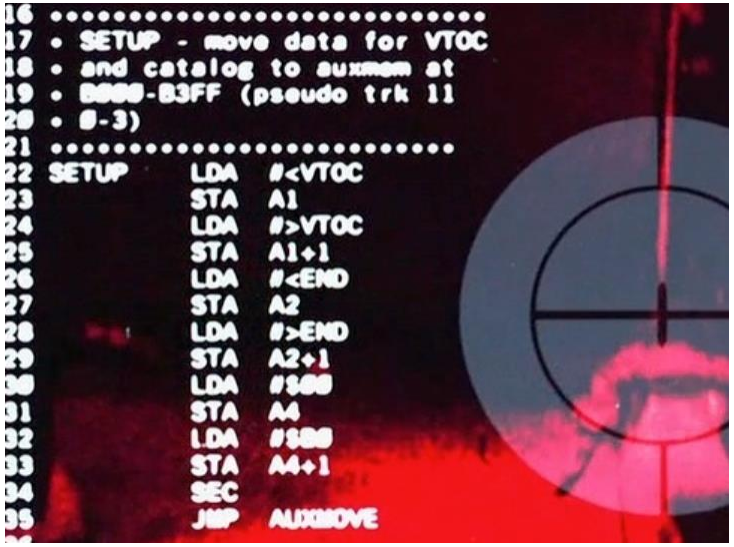


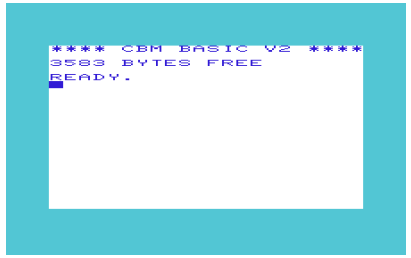
Pizza ordering – invalid first name must be max of 25 characters

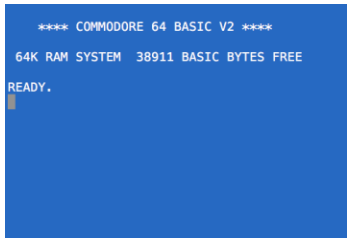


Copyright © 2020 Demetrix, Inc. All rights reserved.

Brief Bio







THE C PROGRAMMING LANGUAGE

python™

DEMETRIX

MONASH University

wellcome sanger institute

EXELIXIS®

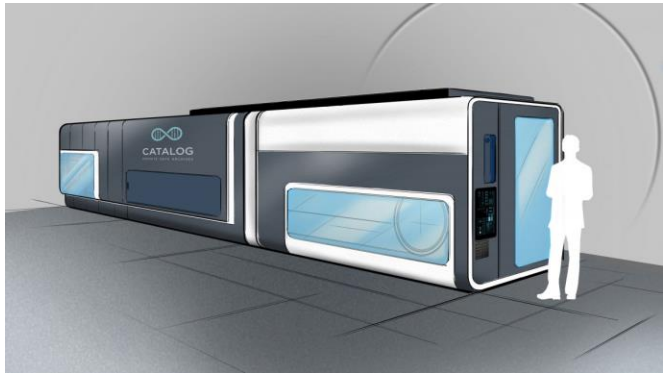
JGI JOINT GENOME INSTITUTE

23andMe

amyris

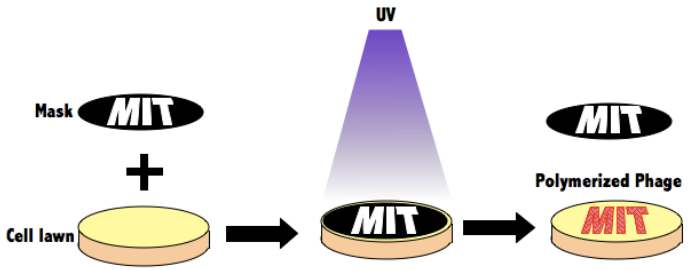
Copyright © 2020 Demetrix, Inc. All rights reserved.

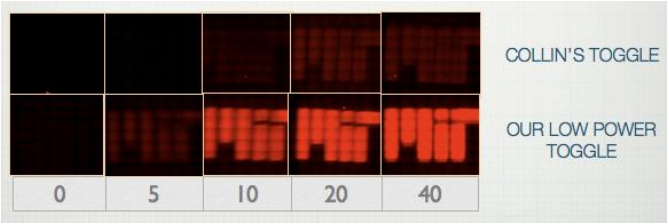
Biotech is cool



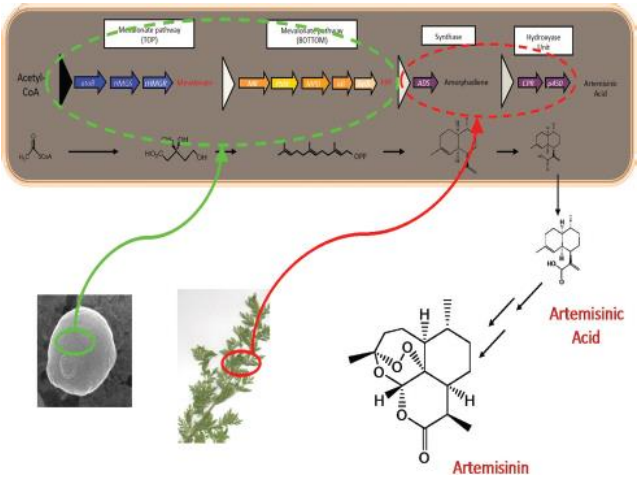
Prototype disk drive writing data into DNA
Catalog DNA

Zbiotics hangover prevention
(bacteria that eats acetaldehyde)

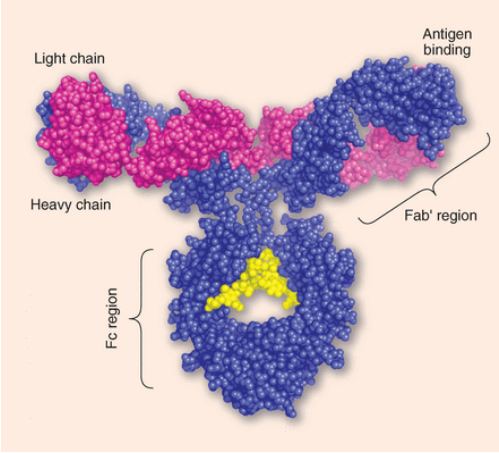




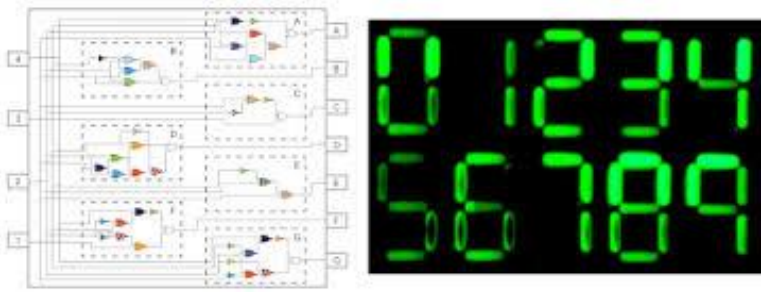
Bacterial camera (**MIT iGEM team**)



Malaria medicine manufactured with yeast
Keasling, Amyris



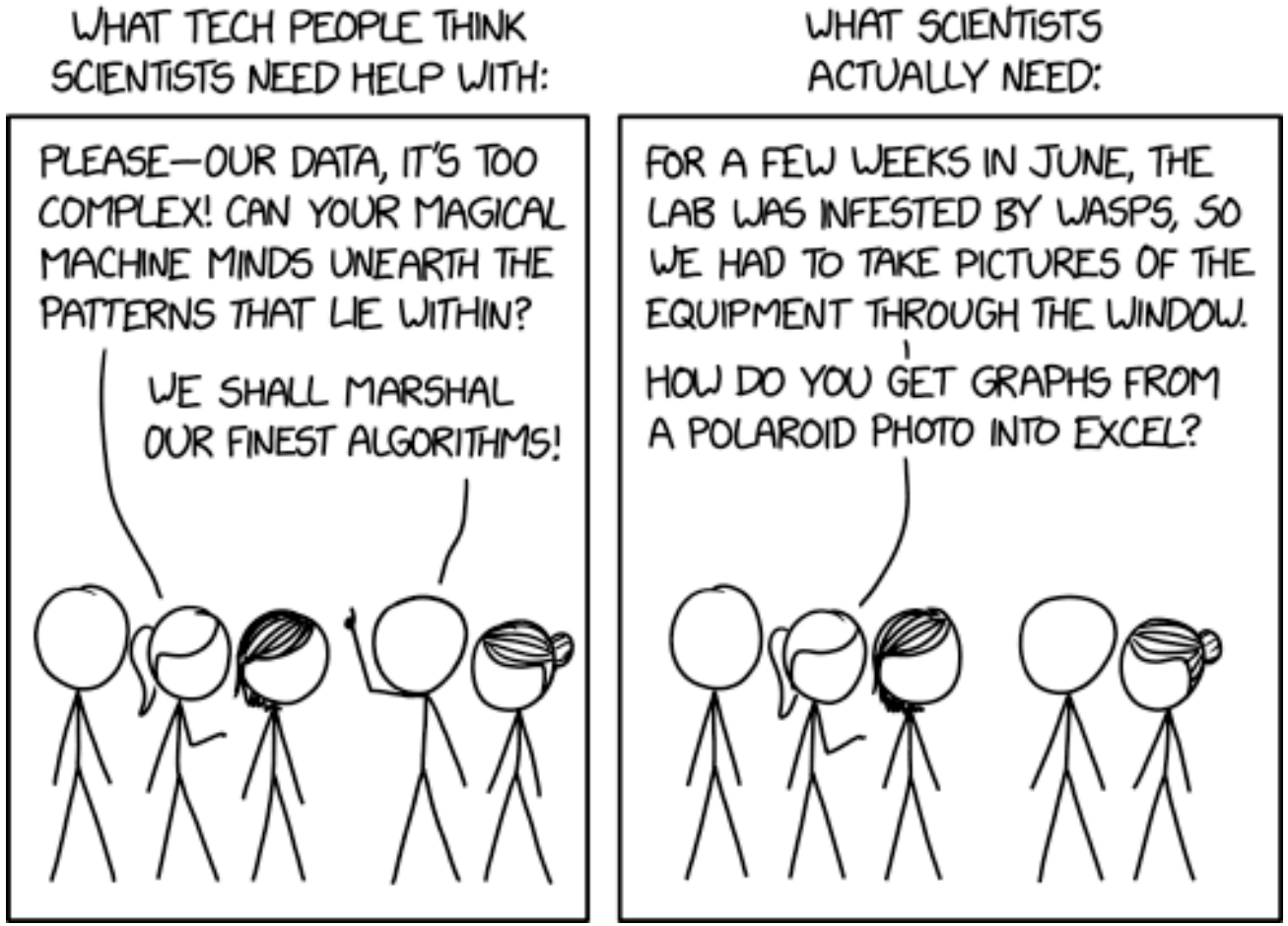
Therapeutic antibodies (cancer , infectious disease)



Bacterial 7 segment display and decoder
MIT Voigt lab

Copyright © 2020 Demetrix, Inc. All rights reserved.

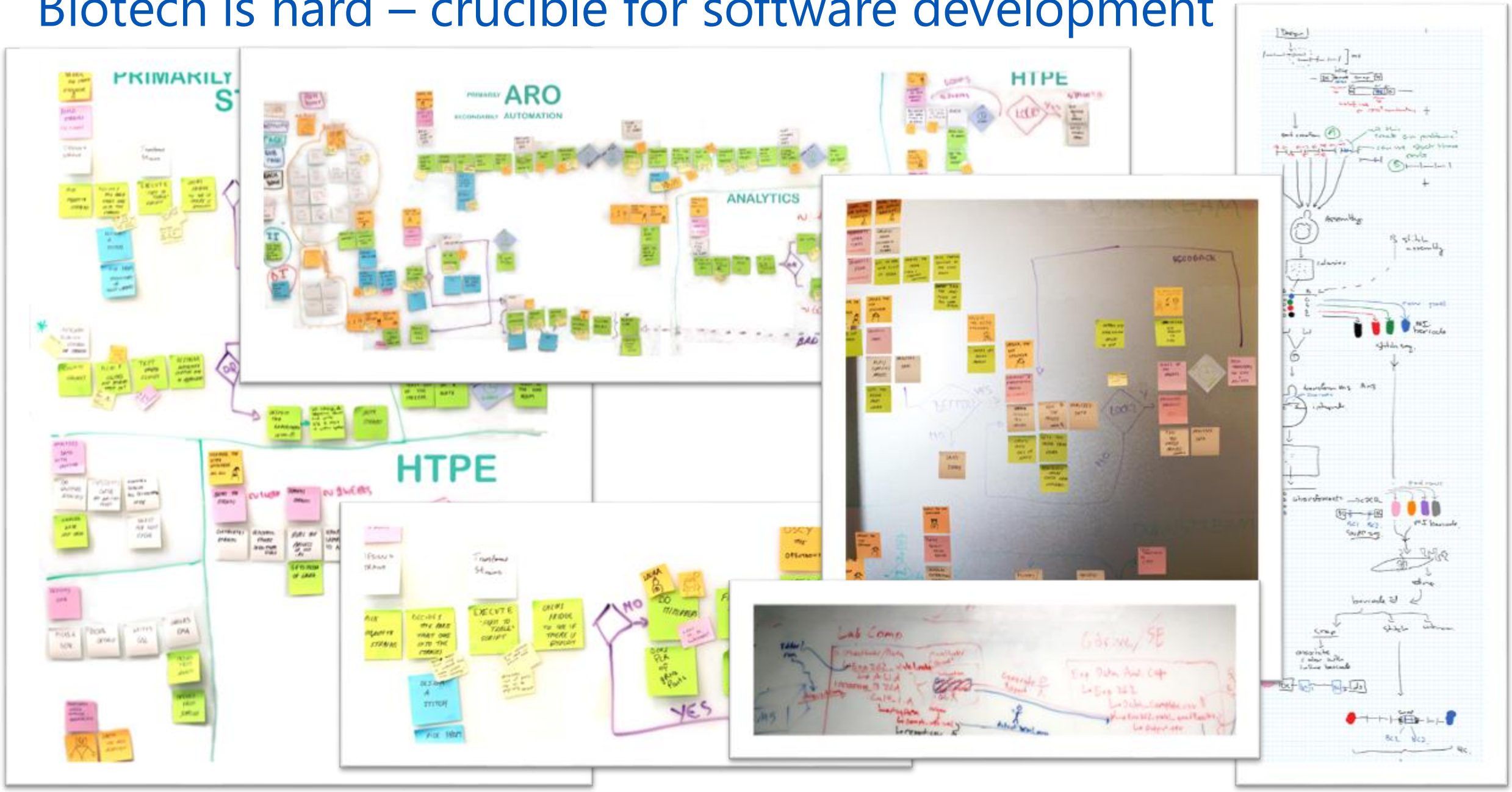
Biotech is seriously misunderstood



https://xkcd.com/2341/

Copyright © 2020 Demetrix, Inc. All rights reserved.

Biotech is hard – crucible for software development



Copyright © 2020 Demetrix, Inc. All rights reserved.

Developers in movies have different problems

DEMETRIX

"If you want to unlock the doors you need to override the security system and enter a new security code
How do I do that ?
You can't .. unless you know BASIC"

(^^^ Really worth watching if you haven't already)

The hacking scene.. BOB cracks the security system

Tricky office environment..

Copyright © 2020 Demetrix, Inc. All rights reserved.

Bob is the only one who knows BASIC

DEMETRIX

```
10 DIM FourDigitPassword INTEGER
20 FOR i = 0 TO 9
30     FOR j = 0 TO 9
40         FOR k = 0 TO 9
50             FOR l = 0 TO 9
60                 FourDigitPassword = getFourDigits (i,j,k,l)
70                 IF checkPasswordMatch(FourDigitPassword) = TRUE THE
80                     GOTO 140
90                 END
100            NEXT l
110        NEXT k
120    NEXT j
130 NEXT i
140 PRINT FourDigitPassword
15_
```

Edgar Dijkstra: Go To Statement Considered Harmful

Go To Statement Considered Harmful

Key Words and Phrases: go to statement, jump instruction, branch instruction, conditional clause, alternative clause, repetitive clause, program intelligibility, program sequencing

dynamic progress is onl
call of the procedure w
we can characterize the
textual indices, the le
dynamic depth of proc

Copyright © 2020 Demetrix, Inc. All rights reserved.

Advantages of F# for Bob's situation

```
seq {for a in 0..9 do
          for b in 0..9 do
               for c in 0..9 do
                    for d in 0..9 -> (a,b,c,d)
} |> Seq.find checkPasswordMatch
```

- Fewer lines! (faster typing)
- No unmaintainable line numbers!
- Avoids gotos
- Type safe solution!
- Friendly scripting environment!

```
> seq {for a in 0..9 do
-           for b in 0..9 do
-                for c in 0..9 do
-                     for d in 0..9 -> (a,b,c,d)
- } |> Seq.find checkPasswordMatch;;
val it : int * int * int * int = (2, 0, 2, 0)
```

Copyright © 2020 Demetrix, Inc. All rights reserved.

[SPOILER ALERT]

Copyright © 2020 Demetrix, Inc. All rights reserved.

Bob is the only one who knows BASIC and dies because of it

Copyright © 2020 Demetrix, Inc. All rights reserved.

Lessons learned

- Choose technology appropriate to problem domain
- Implementation choices do matter
- Considerations
  - Runtimes available in deployment environment
  - Cost of establishing requirements
  - Short term development cost / Developer productivity
  - Software life cycle – life expectancy
  - Long term development cost / Developer productivity
  - Maintenance cost / ability to evolve code base

Copyright © 2020 Demetrix, Inc. All rights reserved.

Goals for this talk

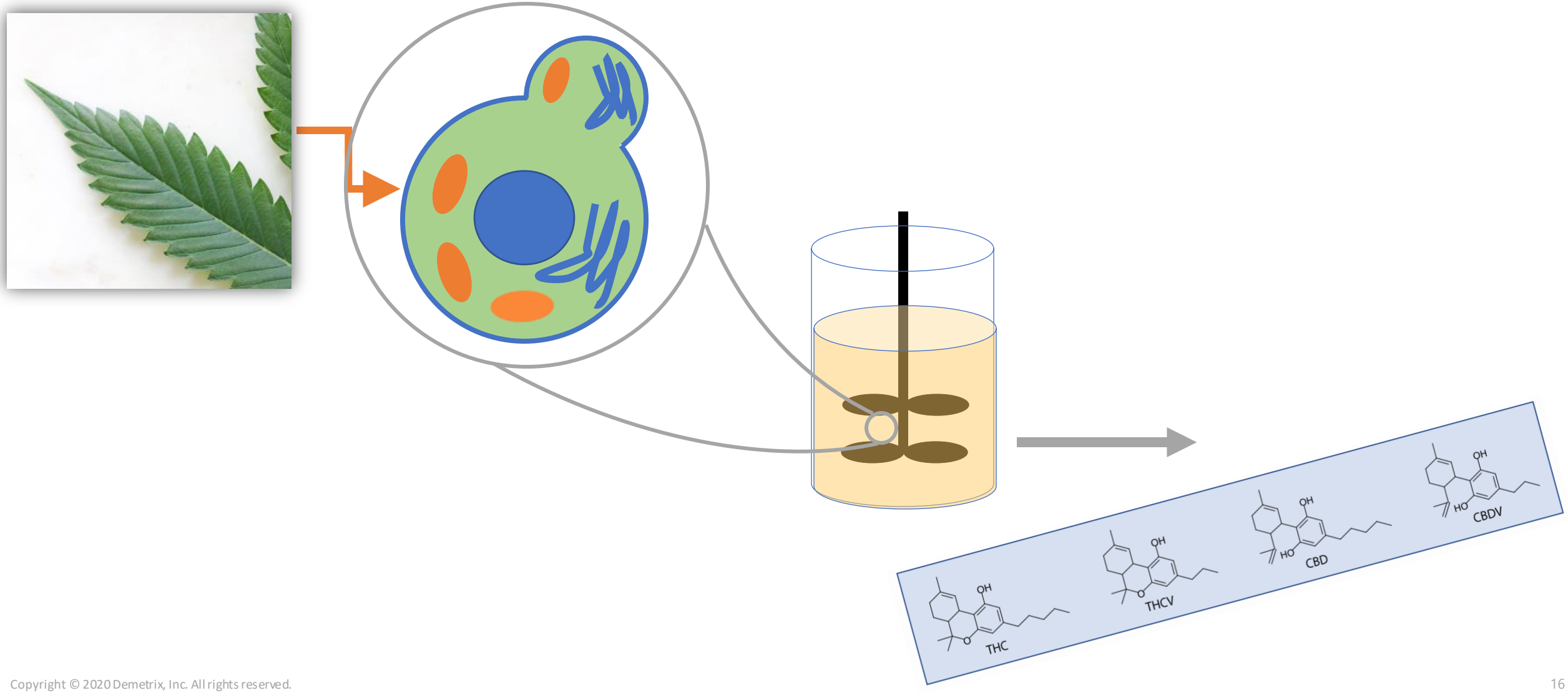
- Crash course in metabolic engineering
- What is unique / common about problems we face
- Why F# is a good fit for those problems
- Examples of software we've built in F# to run a modern lab
- How we use, what we like, obstacles (F# and environment)
- Opportunities aka areas where things could be better

Copyright © 2020 Demetrix, Inc. All rights reserved.

Two minute course on metabolic engineering

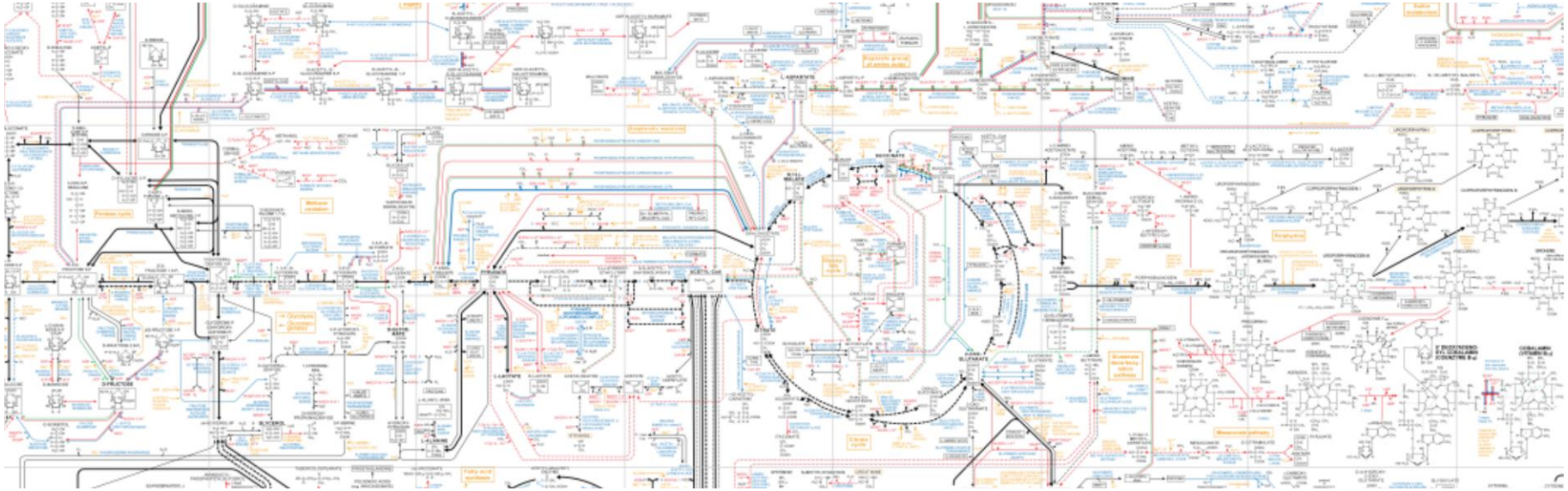
We port plant metabolism into efficient fermentation systems

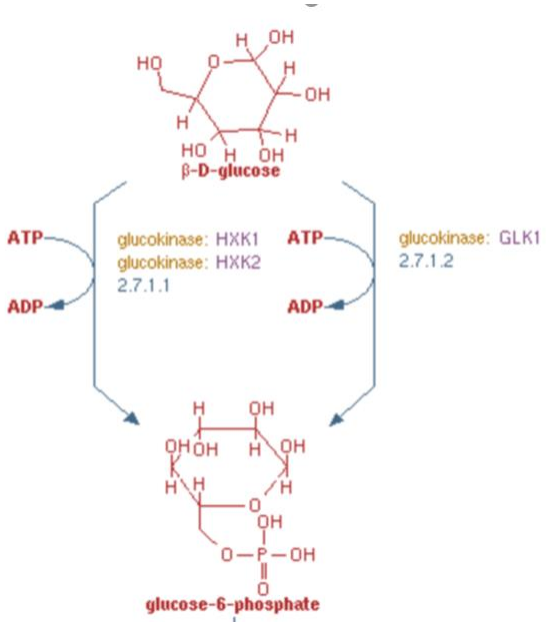
DEMETRIX



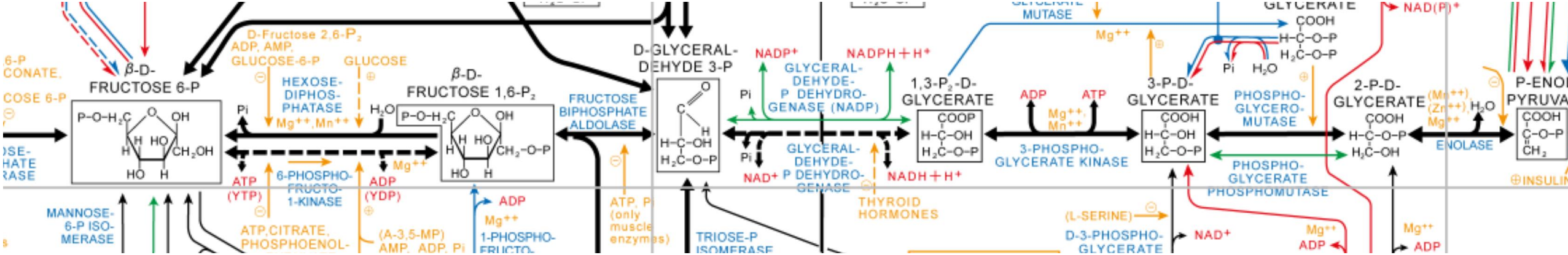
Copyright © 2020 Demetrix, Inc. All rights reserved.

Cells carry out thousands of chemical reactions





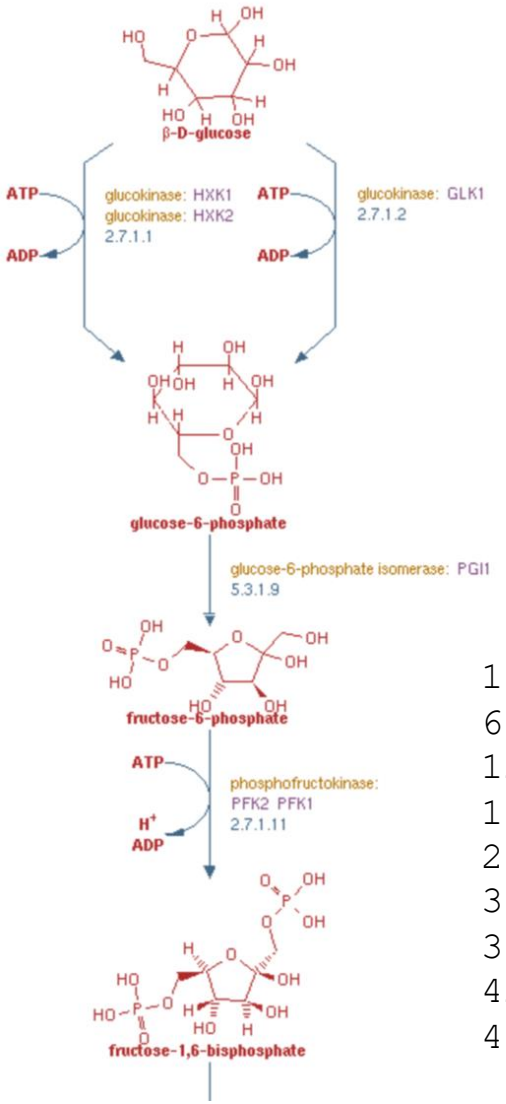
http://biochemical-pathways.com



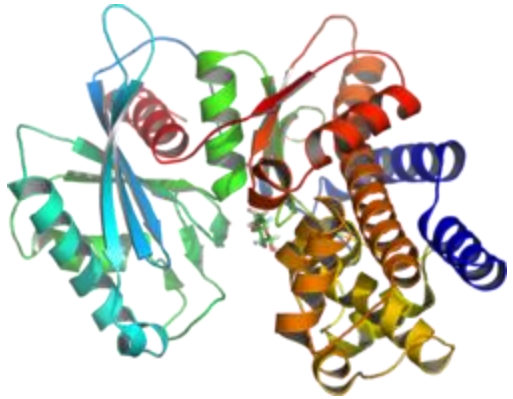
Copyright © 2020 Demetrix, Inc. All rights reserved.

Reactions are carried out by enzymes

DEMETRIX



https://en.wikipedia.org/wiki/Glucokinase

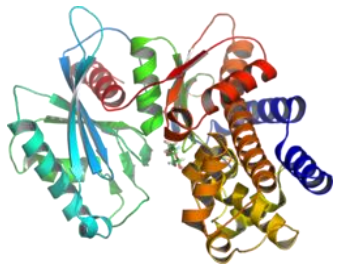


```
1   MVHLGPKKPQ ARKGSMADVP KELMDEIHQL EDMFTVDSET LRKVVKHFID ELNKGLTKKG
61  GNIPMIPGWV MEFPTGKESG NYLAIDLGGT NLRVVLVKLS GNHTFDTTQS KYKLPHDMRT
121 TKHQEELWSF IADSLKDFMV EQELLNTKDT LPLGFTFSYP ASQNKINEGI LQRWTKGFDI
181 PNVEGHDVVP LLQNEISKRE LPIEIVALIN DTVGTLIASY YTDPETKMGV IFGTGVNGAF
241 YDVVSDIEKL EGKLADDIPS NSPMAINCEY GSFDNEHLVL PRTKYDVAVD EQSPRPGQQA
301 FEKMTSGYYL GELLRLVLLE LNEKGLMLKD QDLSKLKQPY IMDTSYPARI EDDPFENLED
361 TDDIFQKDFG VKTTLPERKL IRRLCELIGT RAARLAVCGI AAICQKRGYK TGHIAADGSV
421 YNKYPGFKEA AAKGLRDIYG WTGDASKDPI TIVPAEDGSG AGAAVIAALS EKRIAEGKSL
481 GIIGA
```

Copyright © 2020 Demetrix, Inc. All rights reserved.

Instructions to build the Enzyme (which is a protein) are encoded in DNA in a gene

DEMETRIX



enzyme

```
ATGGTTCATTTAGGTCCAAAGAAACCACAGGCTAGAAAGGGTTCCATGGCTGATGTGCCCAAGGAATTGATGGATGAAATTCATCAG
TTGGAAGATATGTTTACAGTTGACAGCGAGACCTTGAGAAAGGTTGTTAAGCACTTTATCGACGAATTGAATAAAGGTTTGACAAAG
AAGGGAGGTAACATTCCAATGATTCCCGGTTGGGTCATGGAATTCCCAACAGGTAAAGAATCTGGTAACTATTTGGCCATTGATTTG
GGTGGTACTAACTTAAGAGTCGTGTTGGTCAAGTTGAGCGGTAACCATACCTTTGACACCACTCAATCCAAGTATAAACTACCACAT
GACATGAGAACCACTAAGCACCAAGAGGAGTTATGGTCCTTTATTGCCGACTCTTTGAAGGACTTTATGGTCGAGCAAGAATTGCTA
AACACCAAGGACACCTTACCATTAGGTTTCACCTTCTCGTACCCAGCTTCCCAAAACAAGATTAACGAAGGTATTTTGCAAAGATGG
ACCAAGGGTTTCGATATTCCAAATGTCGAAGGCCACGATGTCGTCCCATTGCTACAAAACGAAATTTCCAAGAGAGAGTTGCCTATT
GAAATTGTAGCATTGATTAATGATACTGTTGGTACTTTAATTGCCTCATACTACACTGACCCAGAGACTAAGATGGGTGTGATTTTC
GGTACTGGTGTCAACGGTGCTTTCTATGATGTTGTTTCCGATATCGAAAAGTTGGAGGGCAAATTAGCAGACGATATTCCAAGTAAC
TCTCCAATGGCTATCAATTGTGAATATGGTTCCTTCGATAATGAACATTTGGTCTTGCCAAGAACCAAGTACGATGTTGCTGTCGAC
GAACAATCTCCAAGACCTGGTCAACAAGCTTTTGAAAAGATGACCTCCGGTTACTACTTGGGTGAATTGTTGCGTCTAGTGTTACTT
GAATTAAACGAGAAGGGCTTGATGTTGAAGGATCAAGATCTAAGCAAGTTGAAACAACCATACATCATGGATACCTCCTACCCAGCA
AGAATCGAGGATGATCCATTTGAAAACTTGGAAGATACTGATGACATCTTCCAAAAGGACTTTGGTGTCAAGACCACTCTGCCAGAA
CGTAAGTTGATTAGAAGACTTTGTGAATTGATCGGTACCAGAGCTGCTAGATTAGCTGTTTGTGGTATTGCCGCTATTTGCCAAAAG
AGAGGTTACAAGACTGGTCACATTGCCGCTGACGGTTCTGTCTATAACAAATACCCAGGTTTCAAGGAAGCCGCCGCTAAGGGTTTG
AGAGATATCTATGGATGGACTGGTGACGCAAGCAAAGATCCAATTACGATTGTTCCAGCTGAGGATGGTTCAGGTGCAGGTGCTGCT
GTTATTGCTGCATTGTCCGAAAAAAGAATTGCCGAAGGTAAGTCTCTTGGTATCATTGGCGCTTAA
```

DNA (4 letter alphabet)

```
1 MVHLGPKKPQ ARKGSMADVP KELMDEIHQL EDMFTVDSET LRKVVKHFID ELNKGLTKKG
61 GNIPMIPGWV MEFPTGKESG NYLAIDLGGT NLRVVLVKLS GNHTFDTTQS KYKLPHDMRT
121 TKHQEELWSF IADSLKDFMV EQELLNTKDT LPLGFTFSYP ASQNKINEGI LQRWTKGFDI
181 PNVEGHDVVP LLQNEISKRE LPIEIVALIN DTVGTLIASY YTDPETKMGV IFGTGVNGAF
241 YDVVSDIEKL EGKLADDIPS NSPMAINCEY GSFDNEHLVL PRTKYDVAVD EQSPRPGQQA
301 FEKMTSGYYL GELLRLVLLE LNEKGLMLKD QDLSKLKQPY IMDTSYPARI EDDPFENLED
361 TDDIFQKDFG VKTTLPERKL IRRLCELIGT RAARLAVCGI AAICQKRGYK TGHIAADGSV
421 YNKYPGFKEA AAKGLRDIYG WTGDASKDPI TIVPAEDGSG AGAAVIAALS EKRIAEGKSL
481 GIIGA
```

Protein (20 letter alphabet)

Copyright © 2020 Demetrix, Inc. All rights reserved.

Cannabinoid manufacturing via Fermentation

Porting Plant Process into Fermentation

DEMETRIX

Sequence

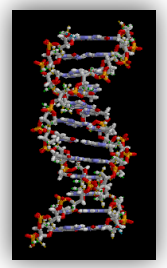
Plant gene

ATGTGACTGATCGATGC
ATGA

Optimized for yeast

ATGTTTAGCTACTGTGAC

synthesis



Transform

Sequence plant, identify key genes

Computational optimization

https://www.nature.com/articles/d41586-019-00714-9

Copyright © 2020 Demetrix, Inc. All rights reserved.

Elucidate Plant Pathway

DEMETRIX

LETTER

https://doi.org/10.1038/s41586

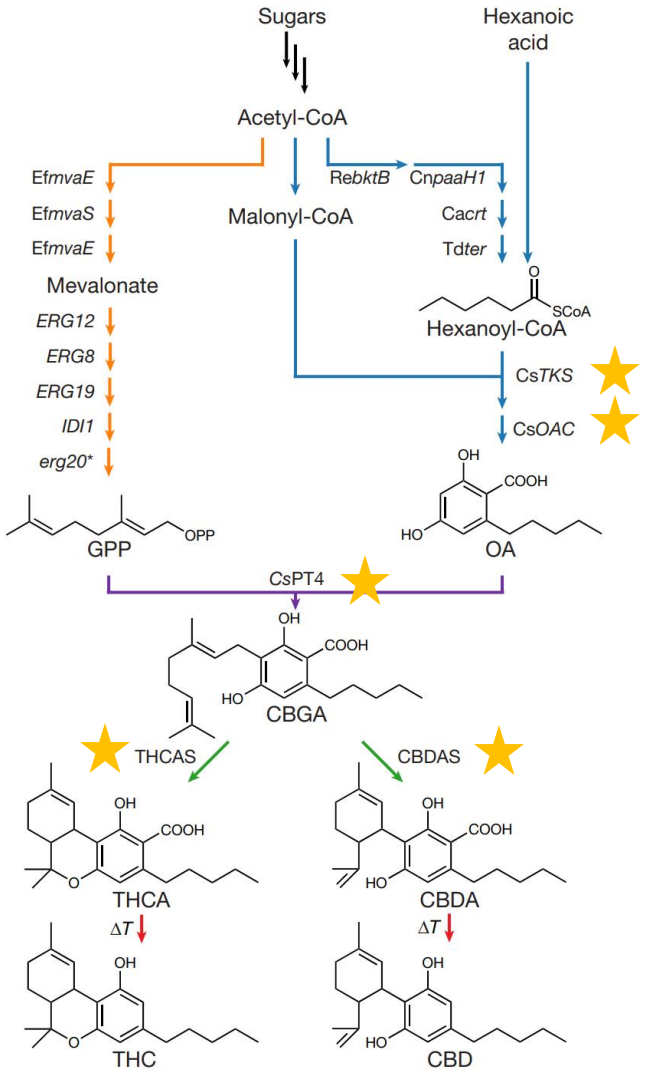
Complete biosynthesis of cannabinoids and thei unnatural analogues in yeast

Xiaozhou Luo[1,15], Michael A. Reiter[1,2,15], Leo d'Espaux[3,12], Jeff Wong[3,12], Charles M. Denby[1,13], Anna Lechner[4,5,14], Yunfeng Zhang[1,6], Adrian T. Grzybowski[1], Simon Harth[3], Weiyin Lin[3], Hyunsu Lee[3,7], Changhua Yu[3,5], John Shin[3,4], Kai Deng[8,9], Veronica T. Benites[3], George Wang[3], Edward E. K. Baidoo[3], Yan Chen[3], Ishaan Dev[3,4], Christopher J. Petzo Jay D. Keasling[1,3,4,5,10,11]*

Cannabis sativa **L. has been cultivated and used around the globe for its medicinal properties for millennia[1]. Some cannabinoids, the hallmark constituents of *Cannabis*, and their analogues have been investigated extensively for their potential medical applications[2]. Certain cannabinoid formulations have been approved as prescription drugs in several countries for the treatment of a range of human ailments[3]. However, the study and medicinal use of cannabinoids has been hampered by the legal scheduling of *Cannabis*, the low in planta abundances of nearly all of the dozens of known cannabinoids[4], and their structural complexity, which limits bulk chemical synthesis. Here we report the complete biosynthesis of the major cannabinoids cannabigerolic**

of hexanoyl-CoA[11]. To increase the supply of hexanoyl-Co yCAN01 strain with 1 mM hexanoic acid, which can be hexanoyl-CoA by an endogenous acyl activating enzyme observed a sixfold increase in olivetolic acid production A known byproduct of TKS, hexanoyl triacetic acid lacto was also detected (Extended Data Fig. 1).

To optimize the conversion of hexanoic acid to hexan introduced *Cs*AAE1, an AAE from *Cannabis* that is thoug this step in planta, into yCAN01[12]. When fed with 1 mM h the resulting strain (yCAN02) showed a twofold increase acid titre (3.0 mg l^{-1}) compared with yCAN01 (Fig. 2a). hexanoyl-CoA from galactose and complete the olivetol



★ *Cannabis sativa* specific enzymatic steps

https://www.nature.com/articles/s41586-019-0978-9

https://demetrixbio.com/media

Copyright © 2020 Demetrix, Inc. All rights reserved.

Scaling Fermentation

DEMETRIX

0.5 liter

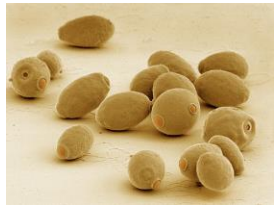
Demetrix lab scale

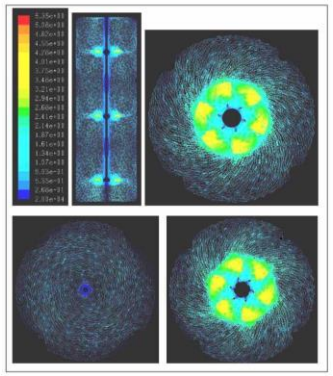
Example 1-2 k liter

https://www.indiamart.com/proddetail/industrial-bioreactors-4275062173.html

Example 200k liter x N

https://cen.acs.org/articles/95/i47/Amyris-sell-farnesene-plant-DSM.html

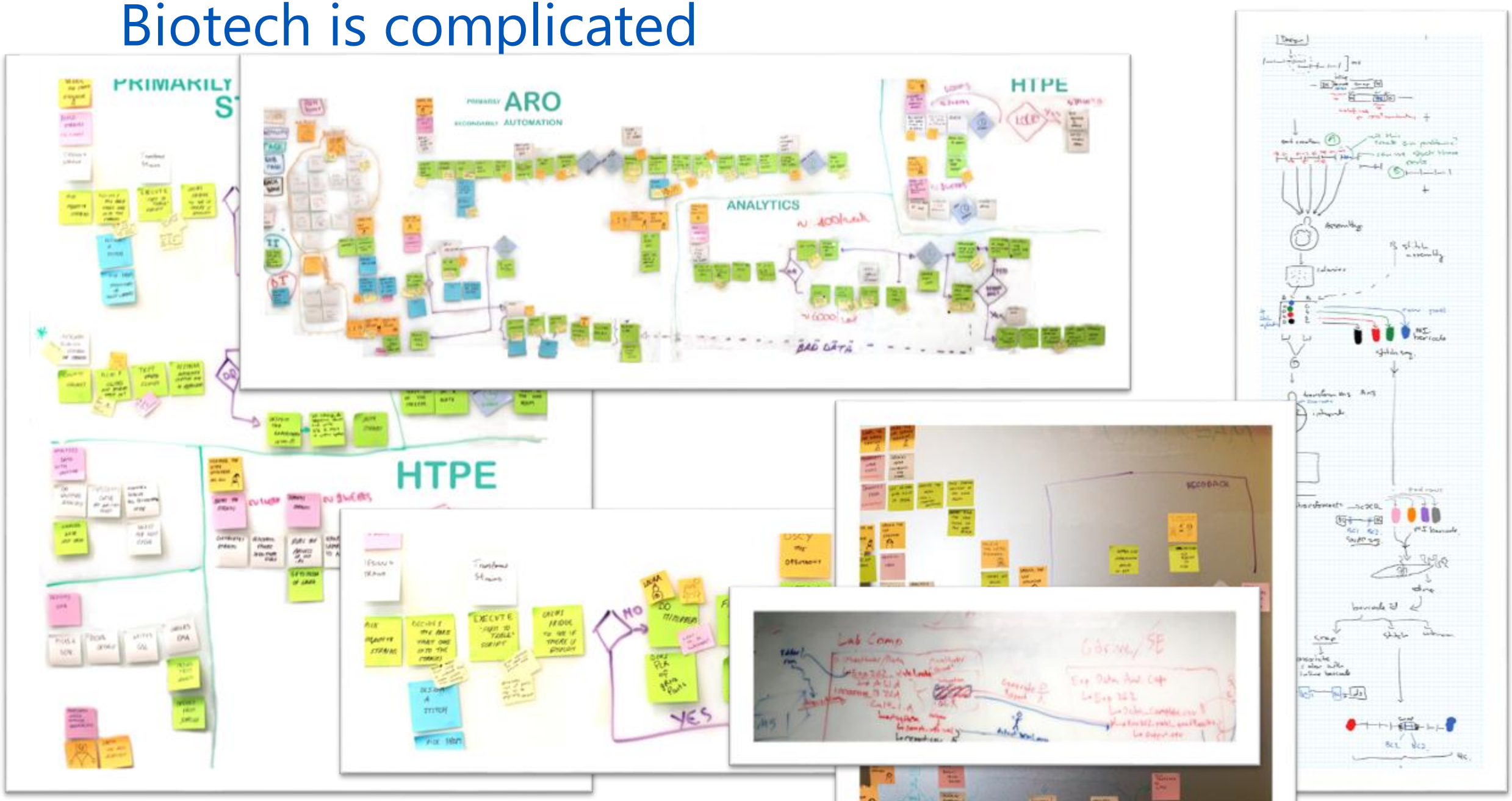




Copyright © 2020 Demetrix, Inc. All rights reserved.

Running a modern lab is hard

Biotech is complicated



Copyright © 2020 Demetrix, Inc. All rights reserved.

Excel-gate

BBC | Sign in | Home | News | Sport | Reel | Worklife | Travel

NEWS

Home | US Election | Coronavirus | Video | World | US & Canada | UK | Business | Tech | Science | Stories

Tech

Excel: Why using Microsoft's tool caused Covid-19 results to be lost

By Leo Kelion
Technology desk editor

1 day ago

Coronavirus pandemic

The badly thought-out use of Microsoft's Excel software was the reason nearly 16,000 coronavirus cases went unreported in England.

And it appears that Public Health England (PHE) was to blame, rather than a third-party contractor.

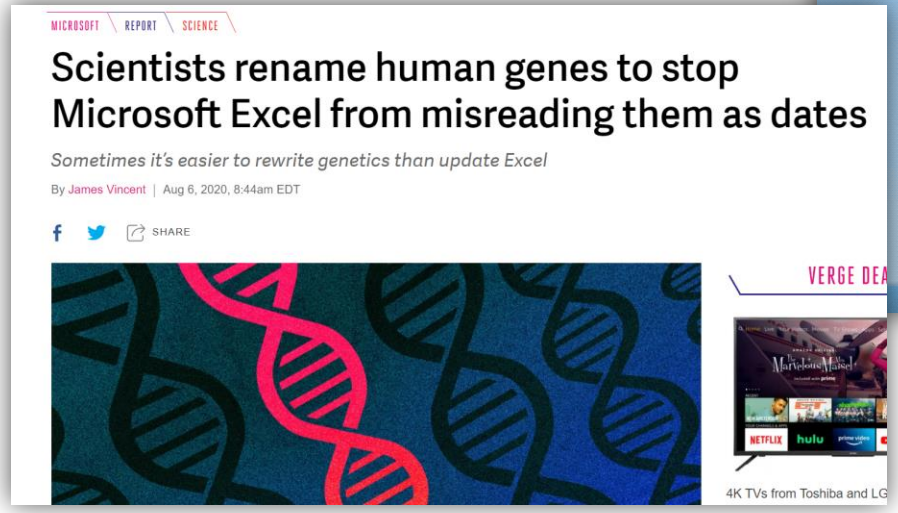
The issue was caused by the way the agency brought together logs produced by commercial firms paid to analyse swab tests of the public, to discover who has the virus.

They filed their results in the form of text-based lists - known as CSV files - without issue.

PHE had set up an automatic process to pull this data together into Excel templates so that it could then be uploaded to a central system and made available to the NHS Test and Trace team, as well as other government computer dashboards.

Mr. Hancock was challenged to put other relevant data-process diagrams into the public domain, so other hidden failings in the government's digital apparatus could be found. But while the minister said he would see what was possible, he added:

"The challenge of a maximum file size error is that it wouldn't necessarily appear on those sorts of flowcharts."



MICROSOFT | REPORT | SCIENCE

Scientists rename human genes to stop Microsoft Excel from misreading them as dates

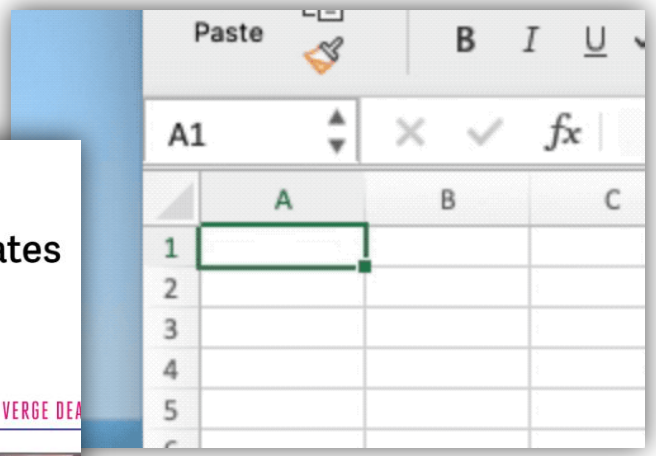
Sometimes it's easier to rewrite genetics than update Excel

By James Vincent | Aug 6, 2020, 8:44am EDT

SHARE

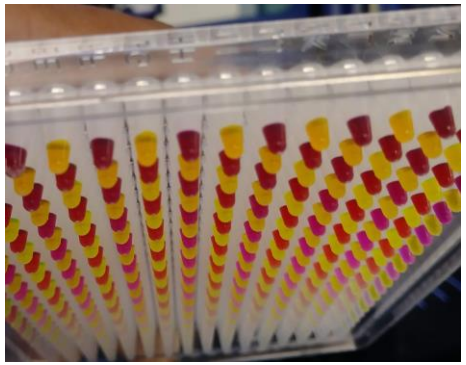
VERGE DEA

4K TVs from Toshiba and LG



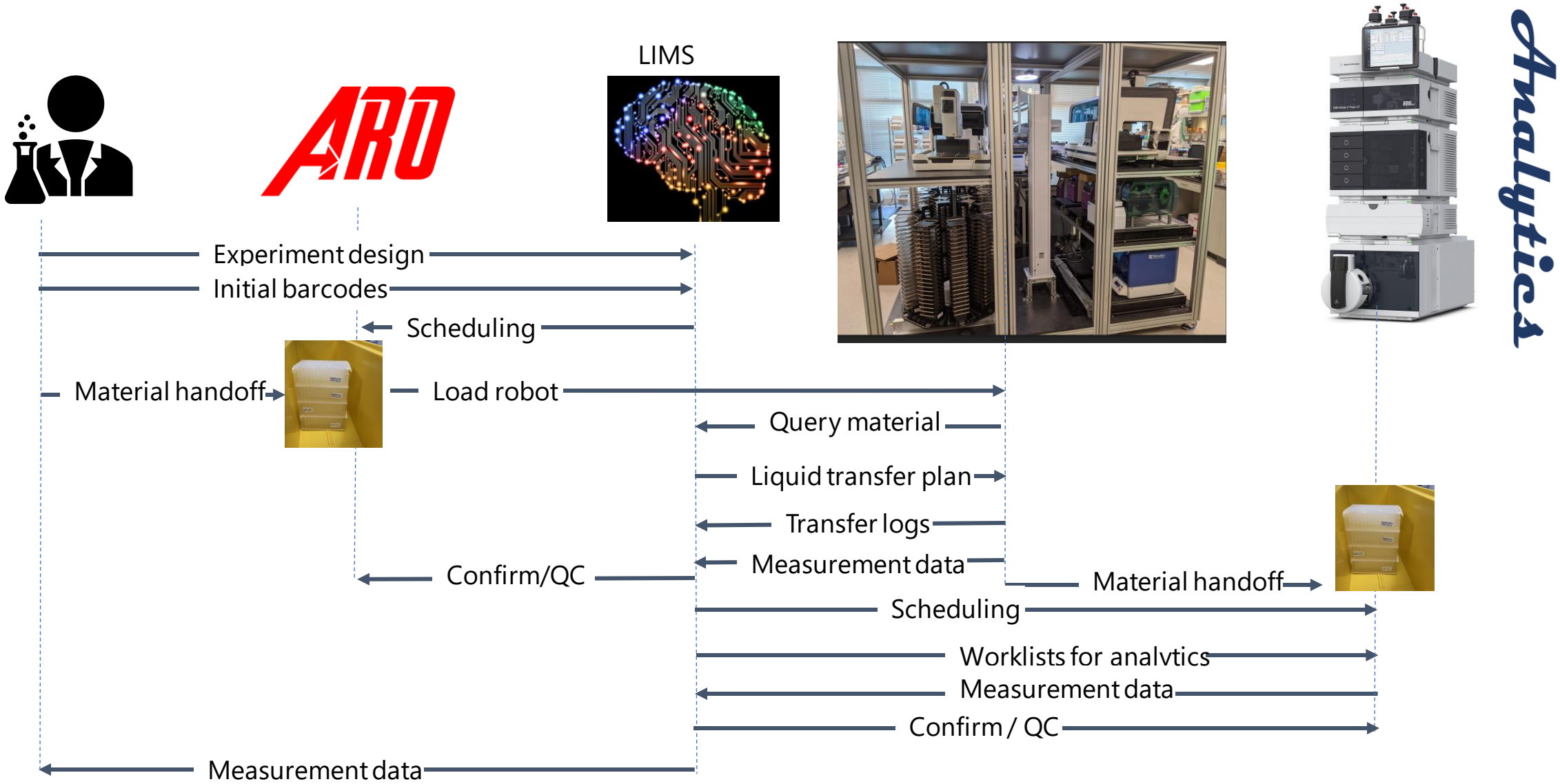
Copyright © 2020 Demetrix, Inc. All rights reserved.

Building a logical mirror of a complex physical world



Copyright © 2020 Demetrix, Inc. All rights reserved.

Laboratory Information Management System



Copyright © 2020 Demetrix, Inc. All rights reserved.

Problem domains

DEMETRIX

- Customer Problem
- Beautiful UI to help person solve problem
- Principled Error Handling
- Secure permissioned access for humans and robots
- Web server technology
- Business logic
- Data storage logic
- Efficient non error prone way of getting data into database
- Reliable data storage technology
- Batch computing solution

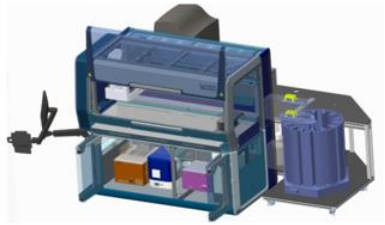
- Requirements Gathering
- Domain modeling
- Prototyping
- MVP
- Testing
- Maintenance
- Iteration

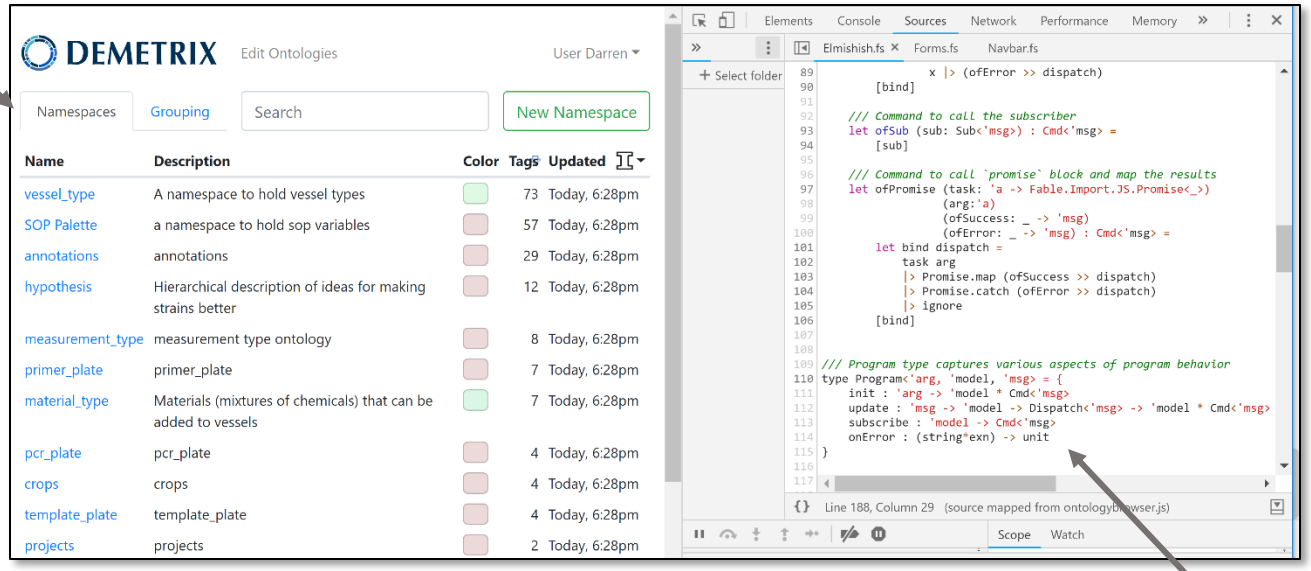
Copyright © 2020 Demetrix, Inc. All rights reserved.

Platform

DEMETRIX

User interfaces
Data visualization
Lab / Robot data services





FABLE

Client

Auth

ASP.NET Core

GIRAFFE

Full stack F#

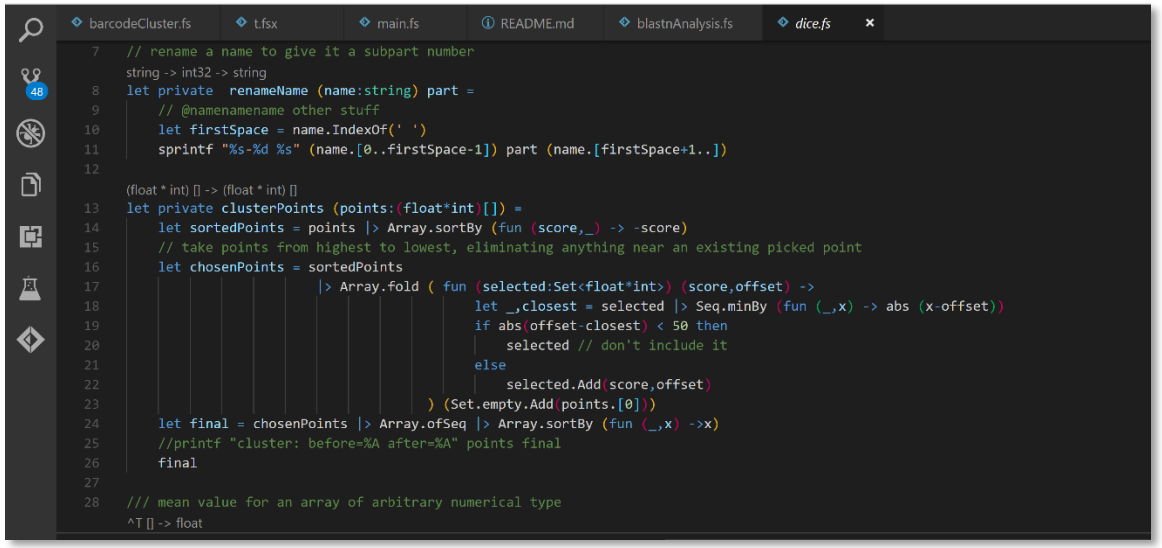
Shared code and data definitions

Hosting

Server

Hangfire
(batch tasks)

Algorithms
DNA Compilers
Data services / access layer

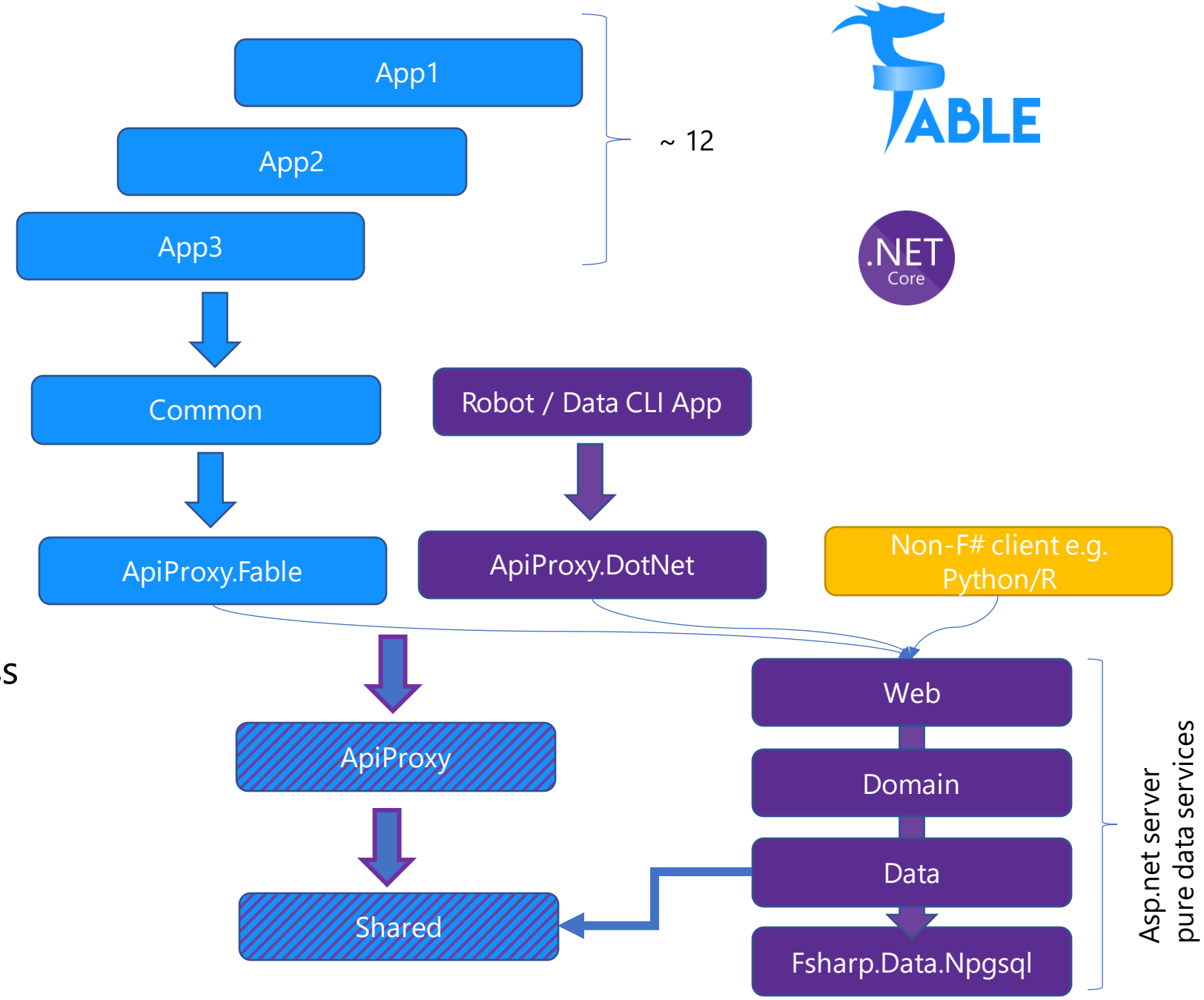


User services
– mail / docs

Copyright © 2020 Demetrix, Inc. All rights reserved.

Code Structure

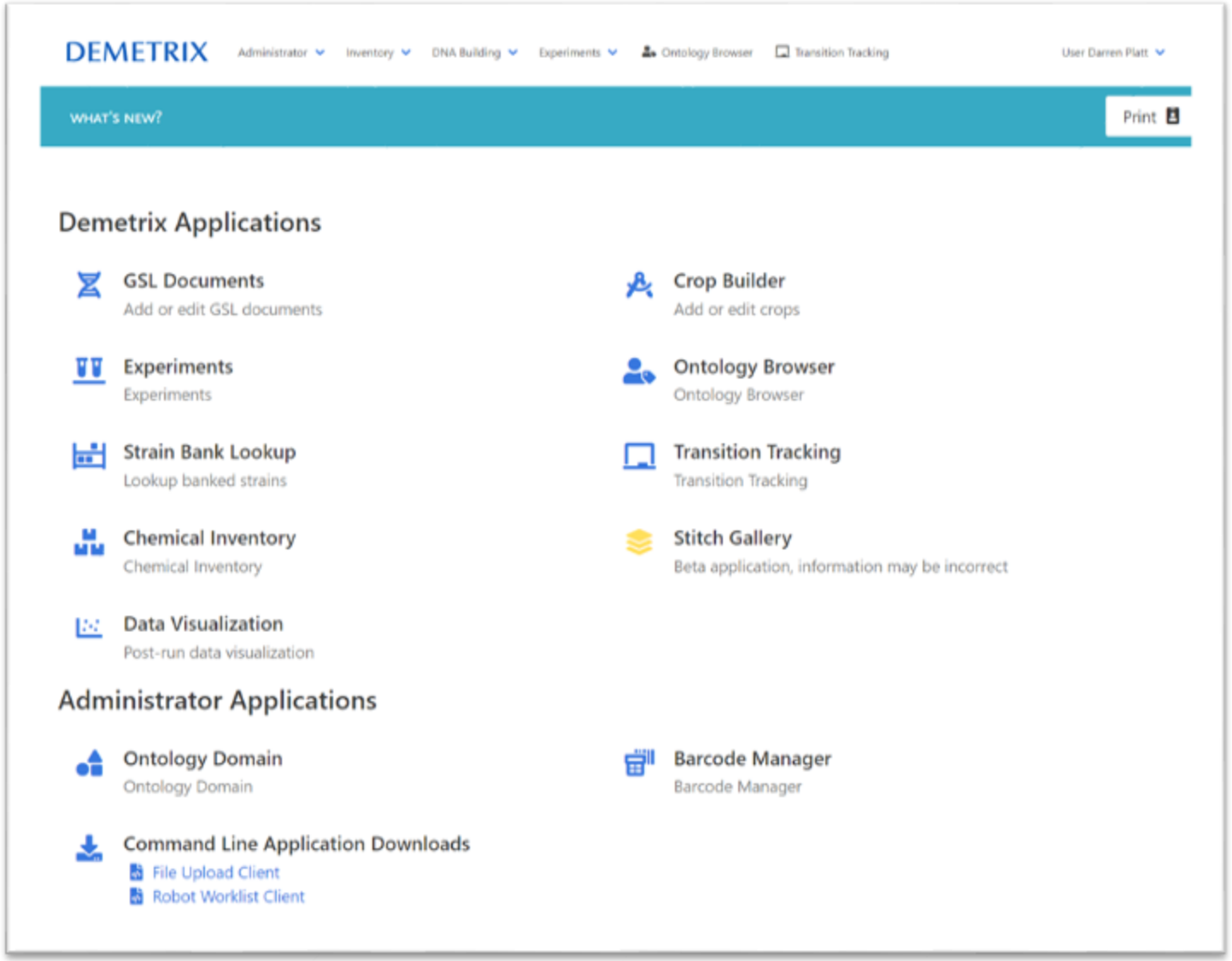
- F# everywhere
- Single data structure definition
- Consistent error handling across front/back end
- Arbitrary app boundaries
  - Consistent Elmish model makes app merge/splits relatively painless
- Command line tools share web api
- Non-F# tooling has equal access to web services
- Single point for access control



Copyright © 2020 Demetrix, Inc. All rights reserved.

Example applications

LIMS is federated set of Fable applications



Copyright © 2020 Demetrix, Inc. All rights reserved.

DNA Design environment

Open F# 2017
https://github.com/demetrixbio
Presentations / Code

DEMETRIX GSL Docs gsl paper demo [copy] User Darren

#49 gsl paper demo

```
// Yeast terpene design from Paddon et al,  Wilson et al.
// Level 1 Syntax
// Part definitions
#refgenome cenpk
let pGAL1_10 = gGAL1[-668:-1]
let truncHMGR = /ATGG/ {#rabitstart } ; gHMG1[1586:~200E]

let cassette(locus,gene1,gene2) =
    &locus.up ; ### ; !&gene1 ; &pGAL1_10 ; &gene2 ; &locus[-500:~500]
end


// Locus engineering
uERG9 ; ### ; pMET3 ; gERG9[1:~500]
cassette(gLEU2,mERG8,mMVD1)
cassette(gHIS3,mERG10,mERG12)
cassette(gADE1,mIDI1,&truncHMGR)
cassette(gURA3,mERG13,&truncHMGR)
cassette(gTRP1,&truncHMGR,mERG20)
```

2018 09/26 15:59.06 Success!

Flags Save And Compile

Download

Genotype Specification Language

Erin H. Wilson, Shiori Sagawa, James W. Weis, Max G. Schubert, Michael Bissell, Brian Hawthorne, Christopher D Reeves, Jed Dean, and Darren Platt*
Amyris, Inc., 5885 Hollis Street, Suite 100, Emeryville, California 94608, United States

Letter | Published: 10 April 2013

High-level semi-synthetic production of the potent antimalarial artemisinin

C. J. Paddon, P. J. Westfall [...] J. D. Newman

Nature **496**, 528–532 (25 April 2013) | Download Citation

Copyright © 2020 Demetrix, Inc. All rights reserved.

DNA Manufacturing Environment

Compiled design from previous page

DEMETRIX Build Applications Crop Editor: open_fsharp_demo_crop User Darren

#50 open_fsharp_demo_crop

GSL Documents

Search Title

#	Title	Owner
47	Crop8_20180926_Leo	Darren Platt
46	crop 8 control stitches	Darren Platt
45	crop7 final	Darren Platt
39	crop6_control_stitches	Darren Platt

Included designs

#	Title	Owner
48	gsl paper demo	Darren Platt

#50 open_fsharp_demo_crop

Statistics

- 6 Upstreams
- 0 Downstreams
- 7 ORF's
- 52 Primers
- 13 Kernels
- 5 Linkers
- 26 PCRs (of which 26 are unique.)
- 12 Stitches/Stalks
- 6 Megastitches/Plants

Missing templates

Name	Size	Used in stalks
!mERG10	1571	gsl_paper_demo_gHIS3.up____!mERG10_gGAL1[-668S:-1S]_mERG12_gHIS3[-500S:~500S]_2of2
!mERG8	1730	gsl_paper_demo_gLEU2.up____!mERG8_gGAL1[-668S:-1S]_mMVD1_gLEU2[-500S:~500S]_2of2
gHMG1[1586S:~200E]	1872	gsl_paper_demo_gADE1.up____!mIDI1_gGAL1[-668S:-1S]_/ATGG/__rabitstart__gHMG1[1586S:~200E]_gADE1[-500S:~500S]_2of2;gsl_paper_demo_gURA3.up

REVERSEPRIMERS

REV_PLATE_1

	1	2	3	4	5	6	7	8	9	10	11	12
A	uLEU2_REV	gERG9[1S:~500S]_REV	uADE1_REV	uURA3_REV	uERG9_REV	uTRP1_REV	uHIS3_REV	pMET3_REV	gGAL1[-668S:-1S]_REV	gHIS3[-500S:~500S]_REV	gLEU2[-500S:~500S]_REV	g[illegible]00S]_REV
B	gURA3[-500S:~500S]_REV	gADE1[-500S:~500S]_REV	!mIDI1_REV	mERG20_REV	mMVD1_REV	mERG12_REV	!mERG13_REV	!gHMG1[1586S:~200E]_REV	snap_a marker2 of2_REV	snap_a marker2 of2_REV	snap_a marker1 of2_REV	!mERG10_REV
C	!mERG8_REV	gHMG1[1586S:~200E]_REV										
D												
E												
F												
G												

DEMETRIX Build Applications Crop Pipeline: open_fsharp_demo_crop

#50 open_fsharp_demo_crop

Crop summary

GSL Documents

48	gsl paper demo	Darren Platt

History

Today, 4:09pm	Darren Platt	Complete Register Physical Plates for PCR
Today, 4:09pm	Darren Platt	Complete Check In Primers
Today, 4:09pm	Darren Platt	Complete Receive Primers
Today, 4:09pm	Darren Platt	Complete Download Primers

Pipeline progress

- Download Primers
- Receive Primers
- Check In Primers
- Register Physical Plates for PCR
- Download PCR Instructions
- Register Plates

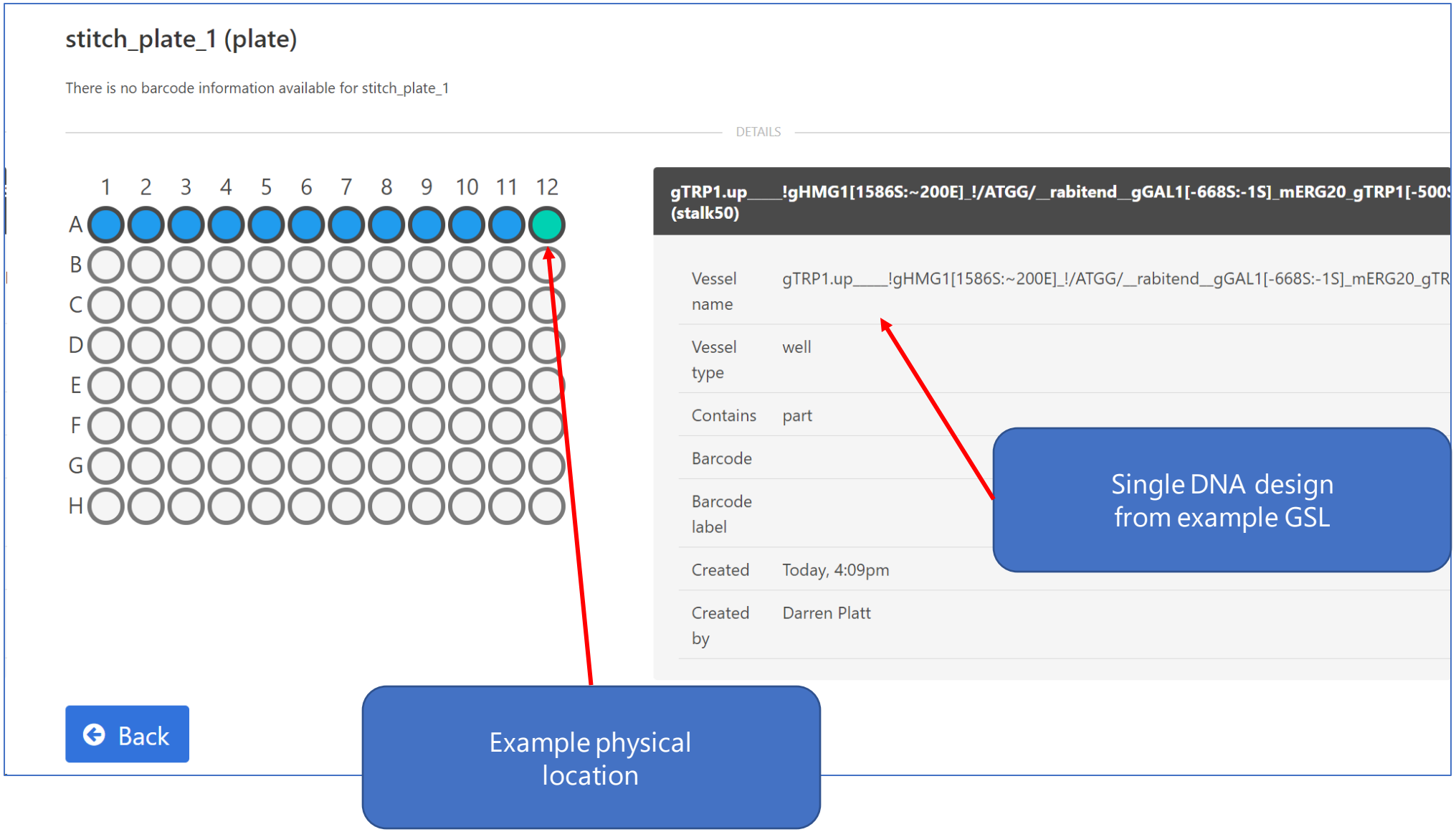
DNA parts needed for construction of design

Copyright © 2020 Demetrix, Inc. All rights reserved.

Building DNA

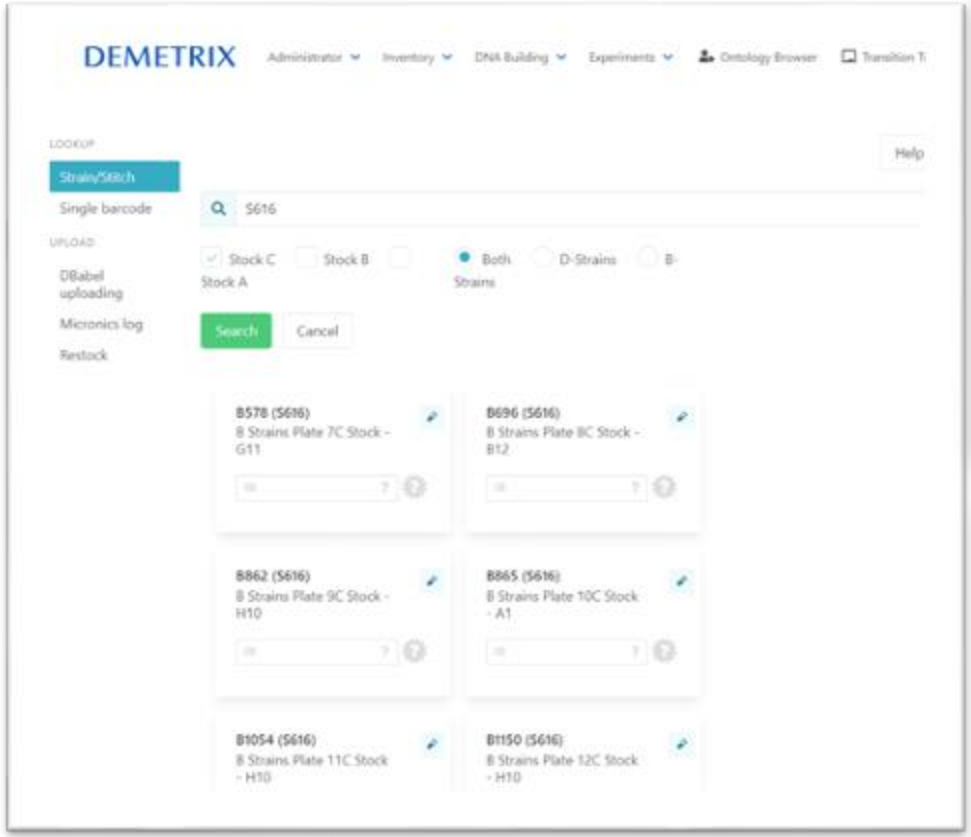
Copyright © 2020 Demetrix, Inc. All rights reserved.

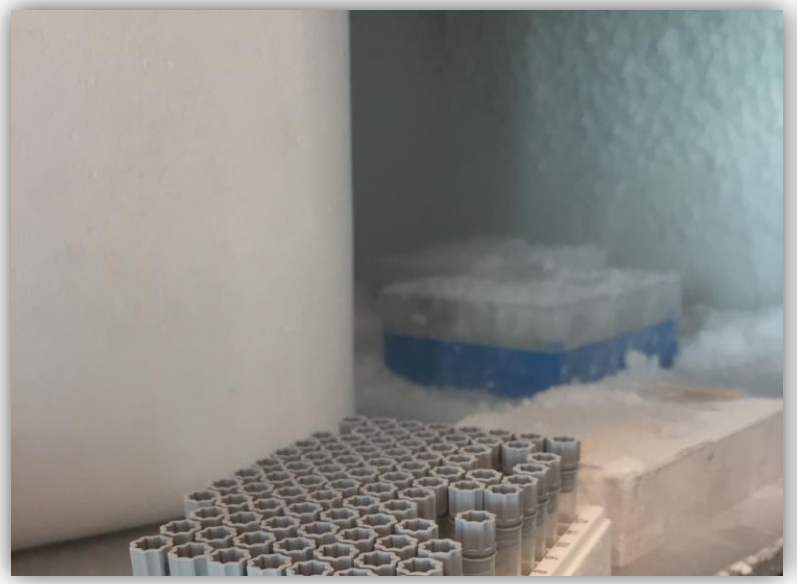
Inventory



Copyright © 2020 Demetrix, Inc. All rights reserved.

Virtual model of Physical World





Copyright © 2020 Demetrix, Inc. All rights reserved.

Coordinating Humans and Robots

Demetrix Services

lims-staging.demetrixbio.net/experimentplanner/

DEMETRIX

Administrator | Inventory | DNA Building | Experiments | Ontology Browser | Transition Trackin

Experiment planner / #aro/schedule/2020-10-12

STRAIN ENGINEERING
- Excel Import
- Experiments

PROCESS DEVELOPMENT & ENGINEERING
- Excel Import
- Experiments
- Single Sample Submission

ARO
- Schedule

ANALYTICS
- Sample Log
- Schedule
- Create Run
- View Runs

ARO Schedule

Look up an experiment

Previous | Next | Show finished stages | Show print button for all days

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Oct 12	Oct 13	Oct 14	Oct 15	Yesterday	Today	Oct 18
Oct 19	Oct 20	Oct 21	Oct 22 Hex 1 EXP445	Oct 23	Oct 24	Oct 25

Demetrix Services

lims.demetrixbio.net/experimentplanner/#experiment/plate/460

DEMETRIX

Administrator | Inventory | DNA Building | Experiments | Ontology Browser | Transition Tracking | User Darren Platt

- Experiments
- Experiment Planner

Experiment planner / Plate Experiment 460 Details

STRAIN ENGINEERING
- Excel Import
- Experiments

PROCESS DEVELOPMENT & ENGINEERING
- Excel Import
- Experiments
- Single Sample Submission

ARO
- Schedule

ANALYTICS
- Sample Log
- Schedule
- Create Run
- View Runs

Experiment: EXP550
Description Making trains
Imported on: Oct 14, 05:13pm
Planned start date: Oct 14
Planned end date: Oct 21
Owner Erin Kim
Status: Running
Project: cannabinoids

Infer experiment progress | Abandon experiment

Planned

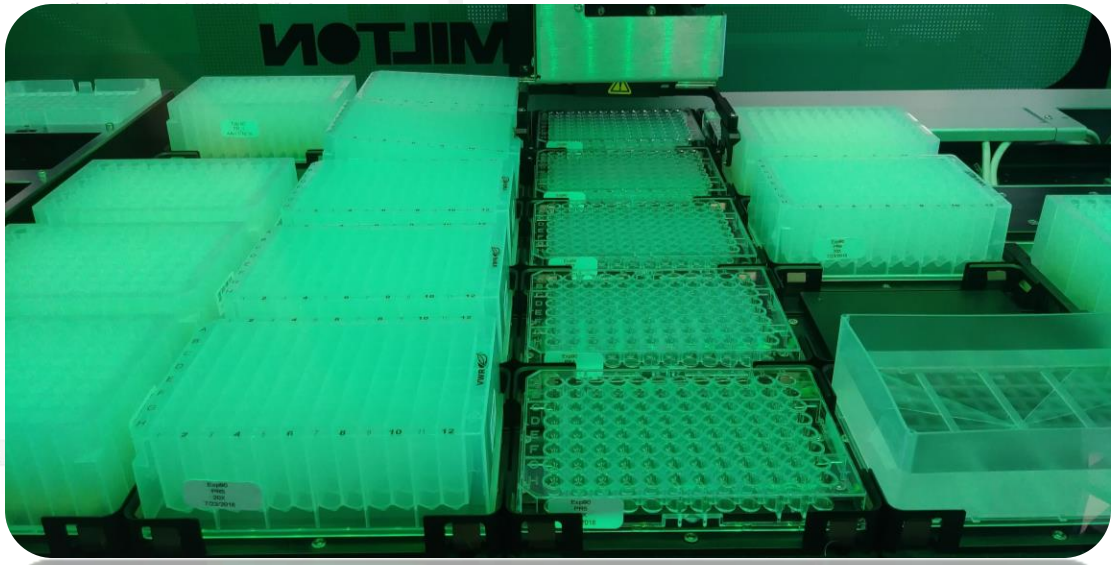
- **PC 1** (Oct 14) - Barcode: **1000042605** - Temp: **30 °C** - Media: **YPD**
  - **PR 1** (Oct 16) - Barcode: **1000040509** - Dilution: **25x** - Temp: **30 °C** - Media: **Jellyfish**
    - **EXT 8** (Oct 18) - Barcode: **--** - Dilution: **40x**
      - **Combined 8** (Oct 18) - Barcode: **--** - Dilution: **25x**
    - **GFP 9** (Oct 18) - Barcode: **--** - Dilution: **1x**
    - **OD 7** (Oct 18) - Barcode: **--** - Dilution: **10x**
  - **PR 3** (Oct 16) - Barcode: **1000040514** - Dilution: **25x** - Temp: **30 °C** - Media: **Jellyfish**
    - **EXT 14** (Oct 21) - Barcode: **--** - Dilution: **40x**
      - **Combined 14** (Oct 21) - Barcode: **--** - Dilution: **25x**
    - **GFP 15** (Oct 21) - Barcode: **--** - Dilution: **1x**
    - **OD 13** (Oct 21) - Barcode: **--** - Dilution: **10x**
  - **GFP 3** (Oct 16) - Barcode: **1000001781** - Dilution: **1x**
  - **Glycerol 2** (Oct 16) - Barcode: **1000042766** - Dilution: **2x**
  - **OD 1** (Oct 16) - Barcode: **1000041092** - Dilution: **10x**
- **PC 2** (Oct 14) - Barcode: **1000042577** - Temp: **30 °C** - Media: **YPD**
  - **PR 2** (Oct 16) - Barcode: **1000040511** - Dilution: **25x** - Temp: **30 °C** - Media: **Jellyfish**
    - **EXT 11** (Oct 18) - Barcode: **--** - Dilution: **40x**
      - **Combined 11** (Oct 18) - Barcode: **--** - Dilution: **25x**
    - **GFP 12** (Oct 18) - Barcode: **--** - Dilution: **1x**
    - **OD 10** (Oct 18) - Barcode: **--** - Dilution: **10x**
  - **PR 4** (Oct 16) - Barcode: **1000040512** - Dilution: **25x** - Temp: **30 °C** - Media: **Jellyfish**
    - **EXT 17** (Oct 21) - Barcode: **--** - Dilution: **40x**
      - **Combined 17** (Oct 21) - Barcode: **--** - Dilution: **25x**
    - **GFP 18** (Oct 21) - Barcode: **--** - Dilution: **1x**
    - **OD 16** (Oct 21) - Barcode: **--** - Dilution: **10x**
  - **GFP 6** (Oct 16) - Barcode: **1000001499** - Dilution: **1x**
  - **Glycerol 5** (Oct 16) - Barcode: **1000040915** - Dilution: **2x**
  - **OD 4** (Oct 16) - Barcode: **1000041101** - Dilution: **10x**

Executed

- **Root** - Barcode: **1000042605**
  - **PR** (Oct 16) - Barcode: **1000040509** - Dilution: **25x** - Media **Jellyfish**
  - **PR** (Oct 16) - Barcode: **1000040514** - Dilution: **25x** - Media **Jellyfish**
  - **GFP** (Oct 16) - Barcode: **1000001781** - Dilution: **--**
  - **Glycerol** (Oct 16) - Barcode: **1000042766** - Dilution: **2x**
  - **OD** (Oct 16) - Barcode: **1000041092** - Dilution: **10x**
- **Root** - Barcode: **1000042577**
  - **PR** (Oct 16) - Barcode: **1000040511** - Dilution: **25x** - Media **Jellyfish**
  - **PR** (Oct 16) - Barcode: **1000040512** - Dilution: **25x** - Media **Jellyfish**
  - **GFP** (Oct 16) - Barcode: **1000001499** - Dilution: **--**

PC

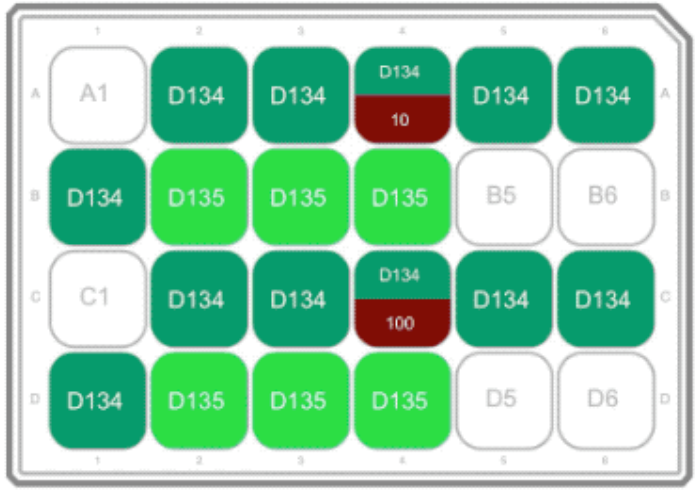
Submission File	Plate	Incubation Length (days)	Temp (°C)
Exp 550.xlsx	Plate: PC1	2	30



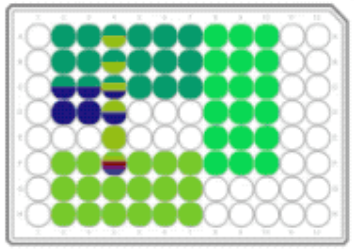
Copyright © 2020 Demetrix, Inc. All rights reserved.

Complex User Interactions

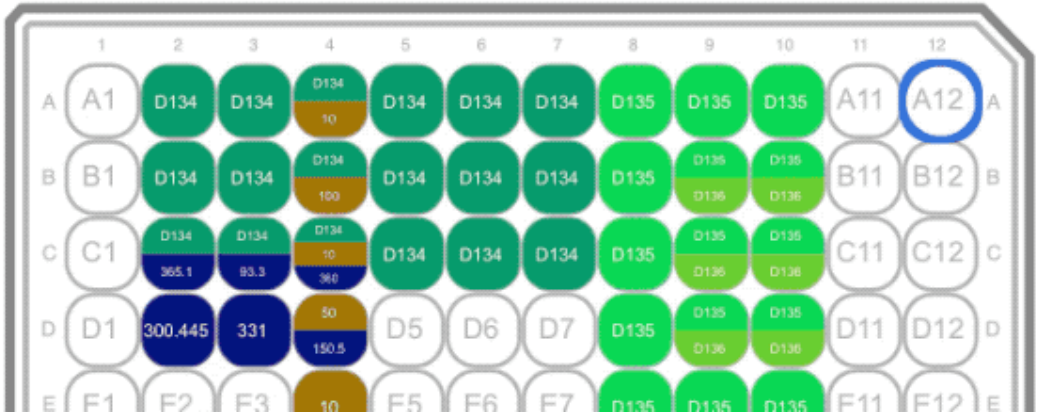
VesselView size Large, 24-well plate



VesselView size Medium, 96-well plate



VesselView size Larger, 96-well plate



Toggle D136

Copyright © 2020 Demetrix, Inc. All rights reserved.

Experimental Data Presentation

DEMETRIX

Administrator | Inventory | DNA Building | Experiments | Ontology Browser | Transition Tracking

Post-run Data Visualization

CHARTS | RAW DATA | CACHE

Equipment

Experiment Types

Run Names

- [ ] PE82-8
- [ ] PE82-9
- [x] PE86-1
- [x] PE86-10
- [x] PE86-11

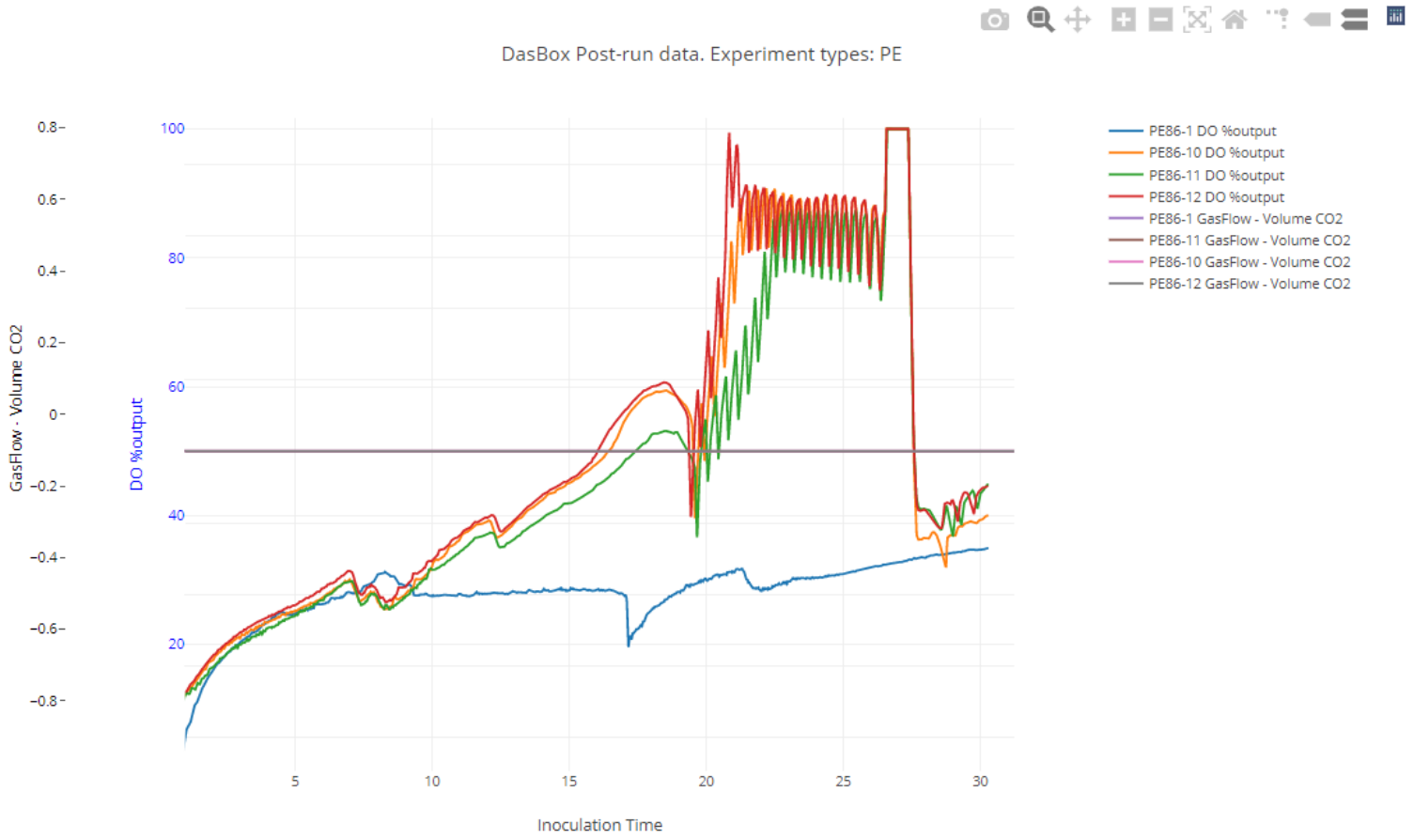
Parameters

- [ ] Agitation SP, [rpm]
- [ ] Agitation, [rpm]
- [x] DO %output, [%]
- [ ] DO SP, [%DO]
- [ ] DO, [%DO]

Apply

Selected experiments:

PE86



Copyright © 2020 Demetrix, Inc. All rights reserved.

Jupyter Notebooks for prototyping / flexible user tools

DEMETRIX

```
In [1]:
// Make sure you're on VPN!
// From toolbar, select "Cell" → "Run All Cells"

// Define input variables here
let expId = 0
let expLabel = "redacted"
let parent = "redacted"
let hitPickType = "RobustZScore" // "PercentImprovement" "PercentImprovementMedian" "RobustZScore" "ZScore"
let hitPickThreshold = 1.0 // for percents, use percent rather than fraction (20.0 instead of 0.2)

let targetBarcodes = [] // Semicolon-separated list of barcodes to transfer hits to
let mutagenesisGDriveFolder = "redacted"

In [3]:
// Does a little bit of  to get vessel types, measurement types, etc but mainly just defines logic

#load "Mutagenesis.fsx"
#load "paket-files/mndrake/IfSharpLab/src/DeedleFormat.fs"
open Deedle

// Query the actual data. Only depends on experiment
let experimentTopology, rawData, rawDataTable = expId ▷ Mutagenesis.getData parent

Requesting access token using client credentials flow from Cognito for app:
Finished loading F# logic
Getting data for Exp  this will take a moment
```

```
In [35]:
Out[35]:
```

	PlateNumber	PCBarcode	WellPosition	...	MeasurementType	Value	MeasurementUnits
0	1	1000031964	A1	...		0.522489147	
1	1	1000031964	A1	...		0.31250536	
2	1	1000031964	A1	...		0	
3	1	1000031964	A1	...		0	
4	1	1000031964	A1	...		0.825000781	
5	1	1000031964	A1	...		0	
6	1	1000031964	A1	...		0.35	
7	1	1000031964	A1	...		0.34	
...	...	...	...	...		...	
782	1	1000031964	H12	...		3.301876814	
783	1	1000031964	H12	...		4.399582434	
784	1	1000031964	H12	...		2.21	
785	1	1000031964	H12	...		3.94	

```
In [5]:
// boxplots
Mutagenesis.plotAllBoxPlots expId hitPickType hitPickThreshold data
// Mutagenesis.plotOneBoxPlot exp hitPickType hitPickThreshold data 1 // this would plot plate number 1
```

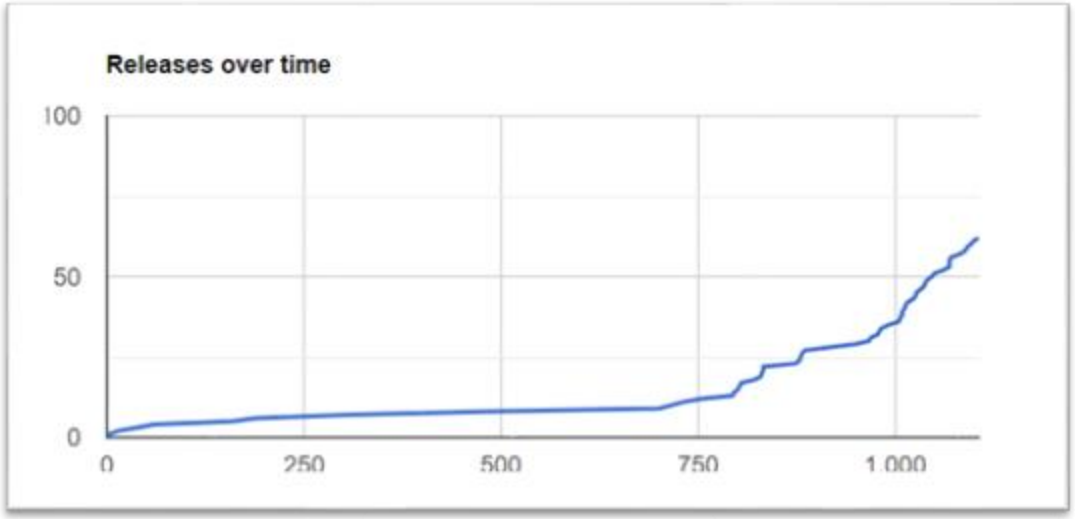
Tooling can leverage full web API for data retrieval

Copyright © 2020 Demetrix, Inc. All rights reserved.

Code

Codebase to support Lab operations

Language	Files	Lines	Code	Comments	Blanks
Batch	4	17	16	0	1
Dockerfile	1	6	6	0	0
F#	651	121625	99628	5614	16383
HTML	3	48	47	0	1
JavaScript	3	523	437	30	56
JSON	20	345	345	0	0
Markdown	10	655	655	0	0
MSBuild	31	1500	1483	12	5
PowerShell	1	12	11	0	1
Python	1	88	52	13	23
Sass	33	2357	1947	56	354
Shell	2	35	24	3	8
SQL	297	14555	11204	1317	2034
SVG	3	521	518	1	2
TypeScript	11	4700	3257	624	819
Visual Studio Sol	1	25	24	0	1
XML	4	1011	1005	4	2
YAML	3	71	47	21	3
Total	1079	148094	120706	7695	19693



SQL
~15k

F#

~54k front end
~49k back end
~10k shared
~5k CLI

"Less, Better, Software™

Copyright © 2020 Demetrix, Inc. All rights reserved.

Elmish Model View Update Example

```
Msg -> Model -> Model * Cmd<Msg>
let update (msg:Msg) model =
    match msg with
    // Renders the empty document browser and initiates a call to fetch documents.
    // Meant to be called once after login.
    | InitCachedResources ->
        model, Cmd.ofAsyncResult CachedResources.init () LoadedCachedResources Oops

    | LoadedCachedResources r ->
        let n = { model.NewExperimentModal with
                    Projects = r.ProjectsSearchData
                    SelectedProject = r.Projects.TagIdsByName.[Projects.Tag.cannabinoids] |> Some
                    ExperimentTypes = r.ExperimentTypesSearchData
                    TagModel = ManageOntologyTags.init "Assign Tags" [r.ExperimentMetadata] Map.empty
        let selectedExperimentType = r.ExperimentTypes.TagIdsByName.[ExperimentType.Tag.Exp] |> Some

        { model with
            CachedResources = Some r
            NewExperimentModal = n
        }, Cmd.ofMsg (NewExperimentModalMsg (ChangeExperimentType selectedExperimentType))

    | Oops exn ->
        model, Toast.fail (sprintf "Something unexpected happened: (%s)." exn.Message)

    //
```

```
table [ClassName tableClassName; Id "hrvDocTable"] [
    thead [] [
        tr [] [
            (tableColHeader "#" "id" BrowserTableColumn.BTCID)
            tableColHeader "Experiment id" "experiment_uid" BrowserTableColumn.BTCExperimentUID
            (
                if model.ShowTags then
                    tableColHeader "Name / Tags" "name" BrowserTableColumn.BTCTitle
                else
                    tableColHeader "Name" "name" BrowserTableColumn.BTCTitle
            )
            (tableColHeader "Description" "description" BrowserTableColumn.BTCDescription)
            (tableColHeader "Owner" "owner" BrowserTableColumn.BTCOwner)
            (tableColHeader "Created" "created" BrowserTableColumn.BTCCreated)
            (tableColHeader "Modified" "modified" BrowserTableColumn.BTCModified)
            (tableColHeader "Stage" "stage" BrowserTableColumn.BTCStage)
            (tableColHeader "Project" "project" BrowserTableColumn.BTCProject)
            th [ Scope "col"; ClassName "dmx-columnchooser" ] [
                div [
                    ClassName "dropdown is-right is-hoverable dmx-columnchooser"
                    Role "menu"
                ] [
                    div [ ClassName "dropdown-trigger" ] [
                        button [ ClassName "button is-white"; AriaHasPopup true; AriaControls "column-
                            span [ ] [ UI.svgIconColumns ]
                            span [ ClassName "icon is-small" ] [
                                i [ ClassName "fas fa-angle-down" ] [ ]
                            ]
                        ]
```

Copyright © 2020 Demetrix, Inc. All rights reserved.

Example web server-side code

DEMETRIX

```
type InstrumentApi(client : LimsHttpClient) =
    InstrumentQuery -> AsyncEither<Instrument list>
    member x.GetInstrumentsByType (query : InstrumentQuery) : AsyncEither<Instrument list> =
        client.Post("/api/harvest_analytics/instruments/query", query)

    InstrumentId -> AsyncEither<Instrument>
    member x.GetInstrumentsById (InstrumentId instrumentId) : AsyncEither<Instrument> =
        client.Get <| sprintf "/api/harvest_analytics/instruments/instrument/%i" instrumentId
```

Thin apiproxy layer

```
// API that requires authenticated user.  Returns boilerplate JSON error message if not authenticated.
routeStartsWith "/api" >=> Handlers.authenticateJSONIfOnline offline >=>
    choose [
        // GSLDocument API (get)

        // Load a source document, summary documents of its parents, and its compiler run
        routef "/api/gsleditor/%i" (makeJSONHandlerWithArg2Async gslAPI.getGSLEditorState)
        // Return a list of summary documents
        route "/api/gsldocumentlist" >=> makeJSONHandler2Async gslAPI.getGSLDocumentSummaries
        routef "/api/download/%i/%s"
                (fun (gslDocId,fileType) ->
                    let docs = gslAPI.getOutputFiles gslDocId fileType
                    let fileName = sprintf "gsl_output_%s.zip" (System.DateTime.Now.ToString("yyyyMMddHHmmss"))
                    let contentType = "application/zip"
                    makeDownload fileName contentType docs)
        route "/api/gsldocument/overview" >=> makeJSONHandler2 gslAPI.getGSLDocumentOverview

        // Ontology API (get)

        subRoute "/api/onto" (
            choose [
                route "/namespace/all" >=> makeJSONHandler2Async ontAPI.getNamespacesAndMasterTags
                routef "/namespace/%i" (makeJSONHandlerWithArg2Async ontAPI.getNamespaceAndTags)
                routef "/namespace/name/%s" (makeJSONHandlerWithArg2Async ontAPI.getNamespaceAndTagsByName)
                route "/domain/all" >=> makeJSONHandler2 ontAPI.getOntologyDomains
                routef "/domain/name/%s" (makeJSONHandlerWithArg2Async ontAPI.getOntologyDomainByName)
                routef "/tagassignment/domainnameandobj/%s/%i" (makeJSONHandlerWithArg2Async ontAPI.getAssigned
```

Standard dispatch of endpoints into code with json / auth wrapping

Copyright © 2020 Demetrix, Inc. All rights reserved.

Result type / Either builder

```
type FailureMessage =
    | Unknown of string
    | Database of string
    | Parse of string
    | Validation of string
    | Conflict of string
    | NotFound of string
    | ExceptionFailure of exn

type Success<'TSuccess> =
    { Data : 'TSuccess
      Warnings : string list }

type Either<'a> = Result<Success<'a>, FailureMessage>
type AsyncEither<'a> = Async<Either<'a>>
```

```
asyncEither {
    use! connection = Db.asyncOpenConnection() |> valueAsSucceed

    let unitNumber = sheetName.Replace("TrackData", "") |> int
    let unit = sprintf "Unit %i" unitNumber
    let! tankLabel = asyncEither {
            match constants.FileType with
            | a when a = FileProcessing.Tag.dasGipOnlineData -> return sprintf "Cub %i" unitNumber
            | a when a = FileProcessing.Tag.dasBoxOnlineData -> return sprintf "Kitten %i" unitNumber
            | TagName a -> return! sprintf "Invalid filetype: %s" a |> Validation |> asyncFail
        }
    let! vesselId, name = Common.getOrCreateVirtualVessel tankLabel constants.ExperimentDetails constants.Lookup constants.IsSeedTrain
    let! tankEquipment = EquipmentService.getEquipmentByNameAndParent { Name = name; Parent = constants.RootEquipment.Id }

    use csv = CsvFile.Parse(csvString, separators = ";", hasHeaders = true)
    let! headers =
        asyncEither {
            match csv.Headers with
            | Some a ->
                return a |> Array.filter(fun x ->
                            ["Timestamp"
                             "Duration"
                             sprintf "%s.InoculationTime []" unit ] |> List.contains x |> not) : string []
                        |> Array.filter(fun x -> x.Contains("Offline") |> not)
            | None -> return! sprintf "No headers for %s" unit |> Validation |> asyncFail
        }

    let timestamp =
        csv.SkipWhile(fun row -> row.GetColumn (sprintf "%s.InoculationTime []" unit) = "").Rows : seq<CsvRow>
```

Copyright © 2020 Demetrix, Inc. All rights reserved.

Example Postgres FSharp.Data.Npgsql interaction

```
use conn = openConnection()
use trans = conn.BeginTransaction()

use gslCompilerRunUpsert = new NpgsqlCommand<"
                                INSERT INTO part.gsl_compiler_run
                                    (id_source_doc, compiler_message, exit_code, errors, warnings,created)
                                VALUES
                                    (@doc_id, @compiler_message, @exit_code, @errors, @warnings,now())
                                ON CONFLICT (id_source_doc)
                                DO UPDATE SET
                                    (compiler_message, exit_code, errors, warnings) =
                                    (@compiler_message, @exit_code, @errors, @warnings)
                                WHERE
                                    part.gsl_compiler_run.id_source_doc = @doc_id",DevCs>(conn)
// Create or update the compiler run record
gslCompilerRunUpsert.Execute(
                doc_id =           doc.index,
                compiler_message = doc.compilerMessage,
                exit_code =        doc.exitCode,
                errors =           doc.errors,
                warnings =         doc.warnings
               ) |> ignore
```

Runtime connection

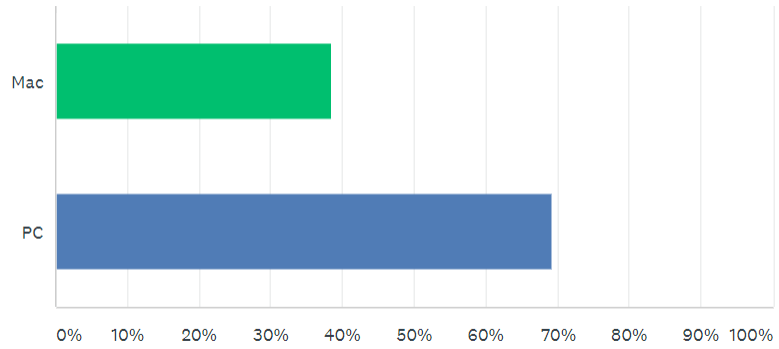
Design time connect string

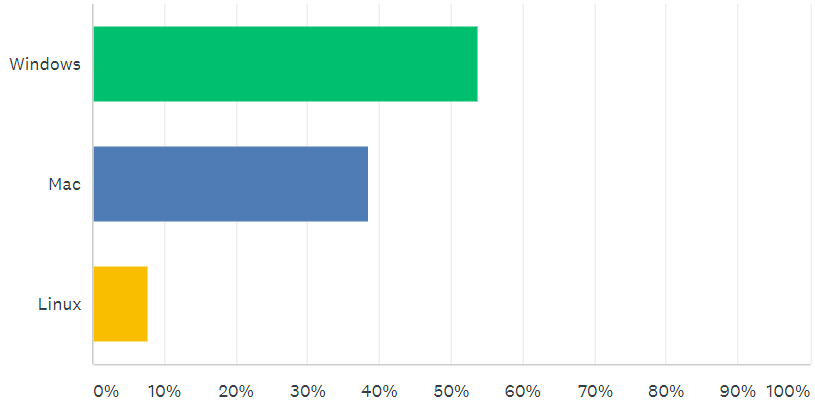
Copyright © 2020 Demetrix, Inc. All rights reserved.

We develop with a healthy mix of different platforms

Project managed as a mono-repo. Typically have whole code base open at once but might selectively watch / recompile for front end development

	Daily	Weekly	Montly	Annually	N/A
Rider	38%	23%	15%	8%	15%
Ionide	31%	15%	23%	8%	23%
VisStudio	0%	0%	0%	20%	80%
Python	15%	15%	23%	15%	31%
R	0%	38%	15%	8%	38%
Jupyter (P)	8%	15%	0%	23%	54%
Jupyter (F#	0%	0%	15%	15%	69%
SQL	46%	38%	15%	0%	0%





Acknowledge a lot of hard work by the tooling community

Copyright © 2020 Demetrix, Inc. All rights reserved.

Paradise? or Room for Improvement?

Functional First not Functional or Die Trying

```
let condProb probB (a:bool []) (b:bool []) =
    let observations = Array.zip a b |> Array.filter ( fun (a,b) -> a&&b) |> Array.length
    {prob = (float observations)/probB/(float a.Length) ; cooccurrence = observations}
    let mutable observations = 0
    let mutable index = 0
    while index < a.Length && index < b.Length do
        if a.[index] = b.[index] then
            observations <- observations + 1
        index <- index + 1
    let observationsF = float observations
    let aLengthF = float a.Length
    let probability = observationsF / probB / aLengthF
    { prob = probability ; cooccurrence = observations}
```

Elegant but slow initial implementation
Running ~1 million of these in parallel

Much faster, imperative replacement optimized for speed
that was slightly incorrect

Copyright © 2020 Demetrix, Inc. All rights reserved.

Rough spots / Opportunities?

- Tooling performance something we watch (and push) – shout out to all the IDE maintainers
- Native JS + Elmish model friction around state
- Scripting seems harder than it should be still (e.g. dependency loading) but that's improving
- Jupyter F# experience still not as good as Python but improving
- Native task state machine support
  - We rely a lot on computational expressions
  - Async / Task heterogeneity
  - Excited about work that's going on to improve this whole area

Copyright © 2020 Demetrix, Inc. All rights reserved.

Stuff we like

- Type providers
  - FSharp.Data – type providers for CSV, XML are huge time savers
  - Fsharp.Data.Npgsql – real time resolution of SQL / type issues (ours)
- Web Server
  - Giraffe / Asp.Net has been reliable and low maintenance
  - Outsourced Auth to AWS
  - Safe stack / paket / FAKE
  - Font awesome
- Data
  - Postgresql / type provider main workhorse
  - Dapper for custom SQL query building
  - Dbup + custom wrapper for migrations
- Web UI
  - Fable
  - Elmish, Fulma , Thoth
  - Xplot, Feliz
- Infrastructure
  - ROP / asyncEither computational builder for routine error handling
  - Hangfire – routine batch computing
  - Nunit for testing
  - Argu
  - FSharp.Collections.ParallelSeq


```
  <DeviceId>HiP-ALS_1</DeviceId>
  <DisplayName>Multisampler</DisplayName>
  <IsRCDevice>True</IsRCDevice>
</Devices>
<Devices>
  <DeviceId>BinPump_1</DeviceId>
  <DisplayName>Binary Pump</DisplayName>
  <IsRCDevice>True</IsRCDevice>
</Devices>
<Devices>
  <DeviceId>QuatPump_1</DeviceId>
  <DisplayName>Quat. Pump</DisplayName>
  <IsRCDevice>True</IsRCDevice>
</Devices>
<Devices>
  <DeviceId>TCC_1</DeviceId>
  <DisplayName>Column Comp.</DisplayName>
  <IsRCDevice>True</IsRCDevice>
</Devices>
<Devices>
  <DeviceId>DAD_1</DeviceId>
  <DisplayName>DAD</DisplayName>
  <IsRCDevice>True</IsRCDevice>
```


(not XML)

Copyright © 2020 Demetrix, Inc. All rights reserved.

There are real monsters out there..

BRIAN BARRETT SECURITY 08.13.19 08:51 PM

HOW A 'NULL' LICENSE PLATE LANDED ONE HACKER IN TICKET HELL

"If anything, he was surprised that the California DMV website let him register NULL in the first place"

"Which means that any time a traffic cop forgot to fill in the license plate number on a citation, the fine automatically got sent to Joseph Tartaro"

"Prank or not, Tartaro was playing with fire by going with NULL in the first place. "He had it coming," says Christopher Null, a journalist who has written previously for WIRED about the challenges his last name presents. "All you ever get is errors and crashes and headaches."

https://www.wired.com/story/null-license-plate-landed-one-hacker-ticket-hell/

Copyright © 2020 Demetrix, Inc. All rights reserved.

Conclusions?

- Biotech is a tough proving ground for a programming environment
- F# has been a great fit for developing software to manage our lab, both back and front end
- Handles complex, changing requirements gracefully, reduces code refactoring-PTSD
- Result is compact, type-safe, reliable code, quickly delivered that is maintainable in face of rapidly evolving requirements

Copyright © 2020 Demetrix, Inc. All rights reserved.

Thank you!

DEMETRIX

Experience the ingredients of life

My minimal shopping list for a programming language

• Strongly typed	- reduces errors
• Semantic whitespace	- reduces errors
• Functional first	- reduces errors
• Option types	- reduces errors
• Units of measure	- reduces errors
• Type inference	- reduces RSI, makes #1 palatable
• Static typing	- fast
• Good parallel primitives	- lazy fast
• Rich built in data types	- reduces development time
• Type providers	- reduces development time
• Open source	- reduces dependency
• Decent library support	- don't want to roll my own db driver
• Language interop (*)	- increases library support
• Mature tooling	- make CPU work for developer
• Supportive community	- life is too short to hang out with jerks

Copyright © 2020 Demetrix, Inc. All rights reserved.