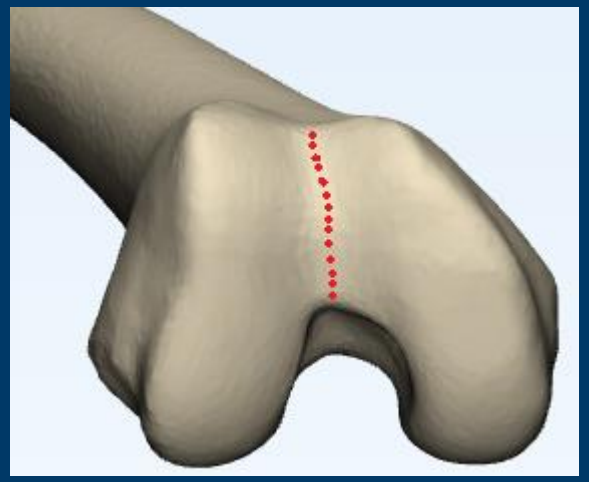
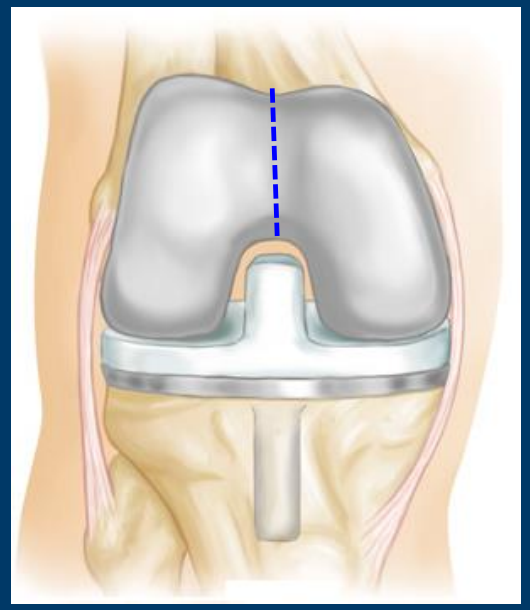
FINDING THE PERFECT FIT

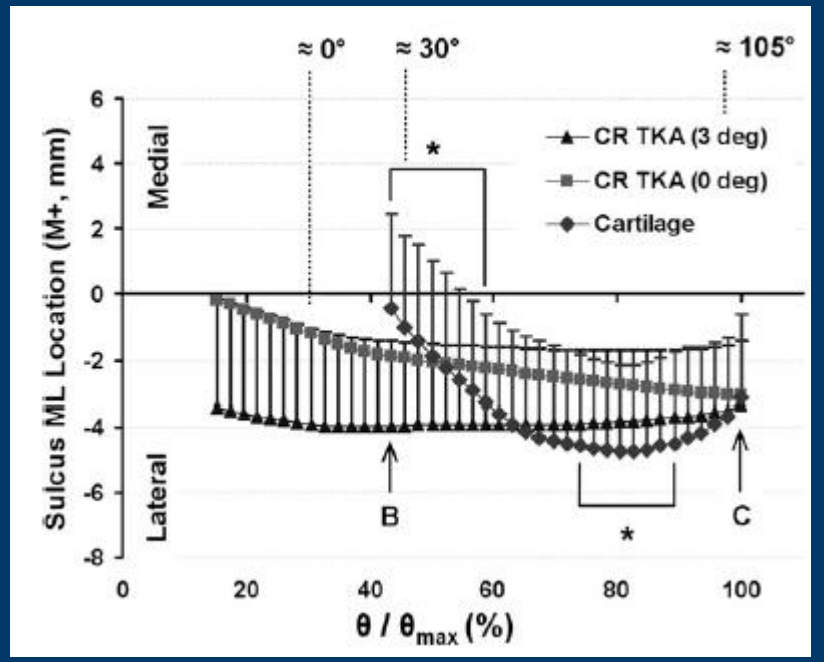
An anatomical population study into the trochlear groove orientation and modern femoral implant designs

Optimized surgical technique ≠ patellofemoral tracking restored to physiological values



VS





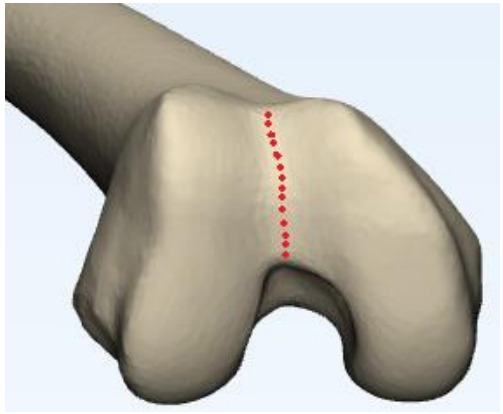
Varadaranjan et al. 2011

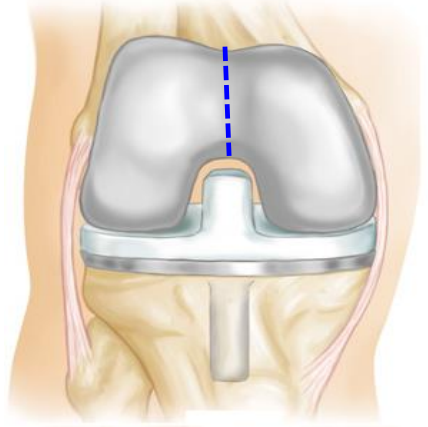
Population Variability

Native Anatomy

Design Philosophy

The “Perfect Fit”

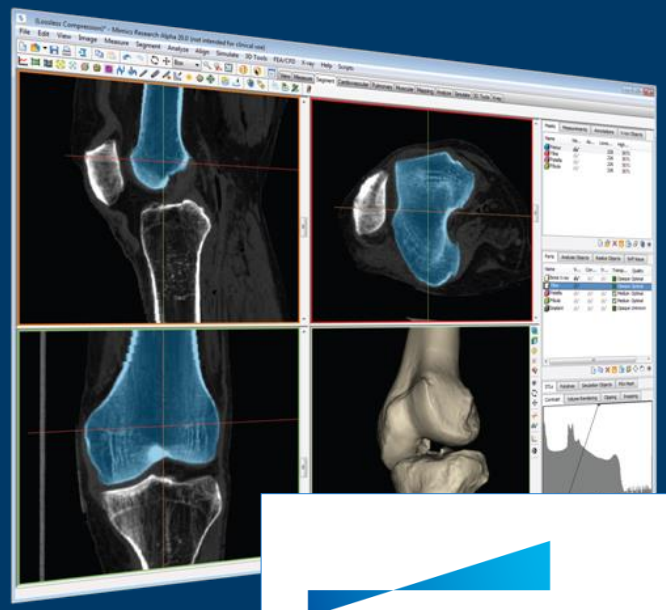




Our Study

Materials

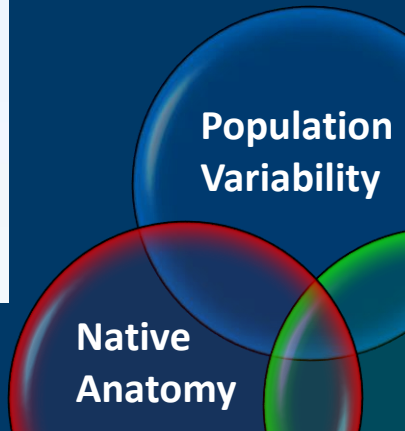
- 100 CT based femoral surfaces
  - 50 Male, 50 Female
  - 50 Asian, 50 Caucasian



```
import time
import numpy as np  # for the math
import numpy.linalg as la  # for the math as well

def compute_angle_signed_on_plane(v1, v2, normal):
    """ Returns the angle in angles between vectors 'v1' and 'v2
    cross = np.cross(v1, v2)
    cosang = np.dot(v1, v2)
    sinang = la.norm(cross)
    angle = np.arctan2(sinang, cosang) * 180 / np.pi
    if np.dot(normal, cross) < 0:
        angle = -angle
    return angle


def angle_signed_on_plane(line1, line2, plane_name):
    normal = trimatic.find_plane(plane_name).normal
    l1 = (line1.get_point(0), line1.get_point(1))
    l2 = (line2.get_point(0), line2.get_point(1))
    v1 = np.subtract(l1[1], l1[0])
    v2 = np.subtract(l2[1], l2[0])
    return compute_angle_signed_on_plane(v1, v2, normal)
```

materialise mimics
innovation suite

Materialise
ADaM



3-matic Scripting

- Rotated a plane around transepicondylar axis in 5° steps
- Generate finely sampled curves representing trochlear region

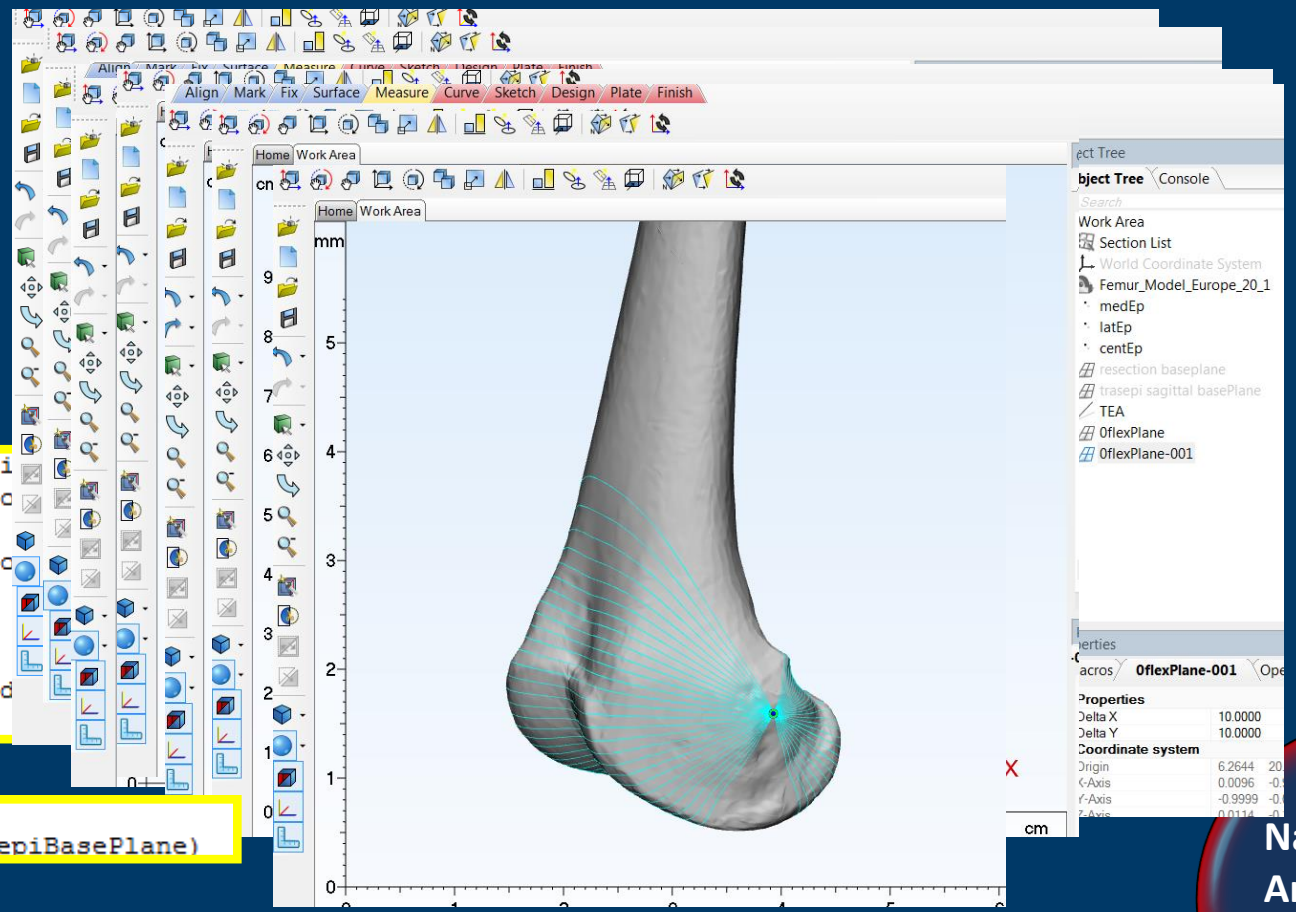
```
import trimatic
import csv
import numpy as np
```

```
for file in os.listdir(path):

    trimatic.new_project()
    trimatic.import_part_stl(file)
    Femur=trimatic.data.parts()[0]
```

```
mpt=trimatic.create_point([mEpix,mEpiy,mEpiz])
lpt=trimatic.create_point([lEpix,lEpiy,lEpiz])
```

```
PlaneNormal = Femur.object_coordinate_system.z_axi
resbasePlane=trimatic.design.create_plane_normal_c
resbasePlane.name='resection baseplane'
sagbasePlane=trimatic.design.create_plane_normal_c
sagbasePlane.name='trasepi sagittal basePlane'

# create 0deg flexion plane
epiBasePlane=trimatic.create_plane_1_point_perpend
epiBasePlane.name='0flexPlane'
```

```
trimatic.rotate(epiBasePlane,5,cpt,TEA,0)
profile=trimatic.create_intersection_curve(Femur,epiBasePlane)
```

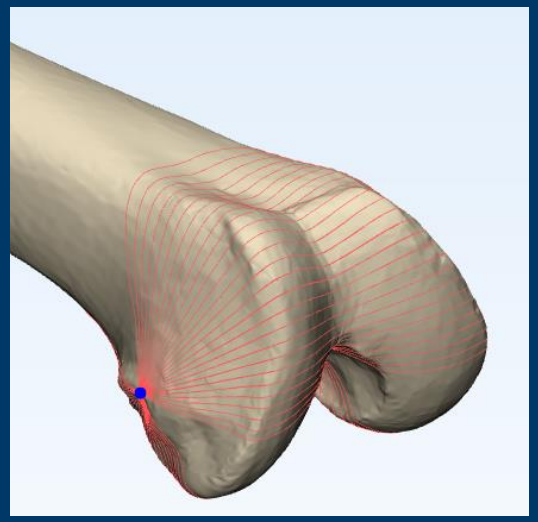


Population Variability

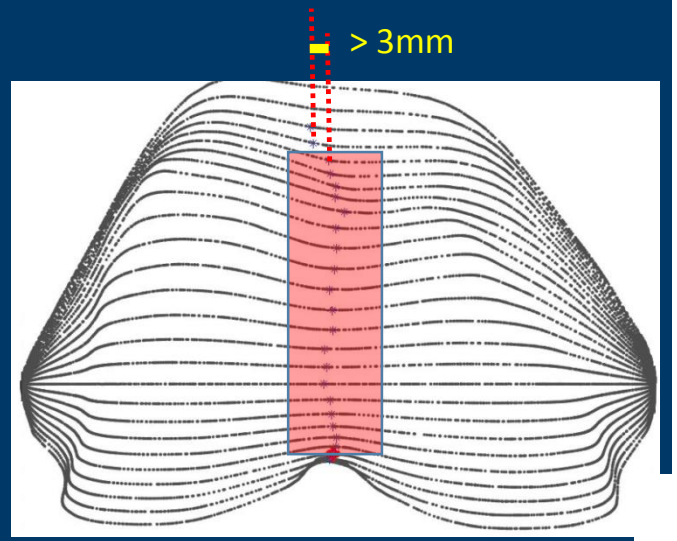
Native Anatomy

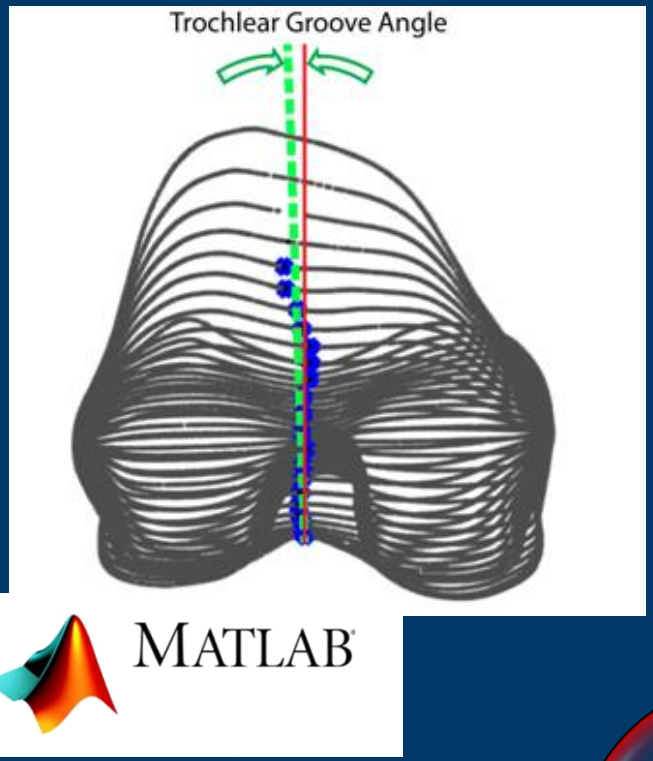
3-matic Scripting

- Deepest points identified on trochlear curves
- Best fitted line for trochlear orientation referencing mechanical axis

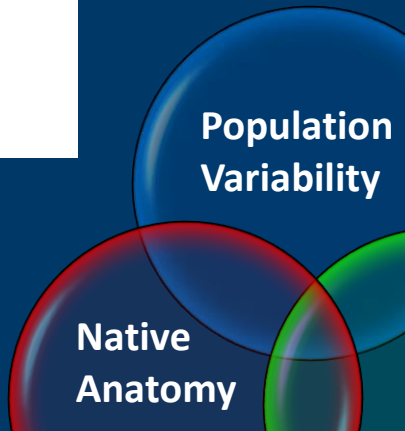


Materialise 3-matic





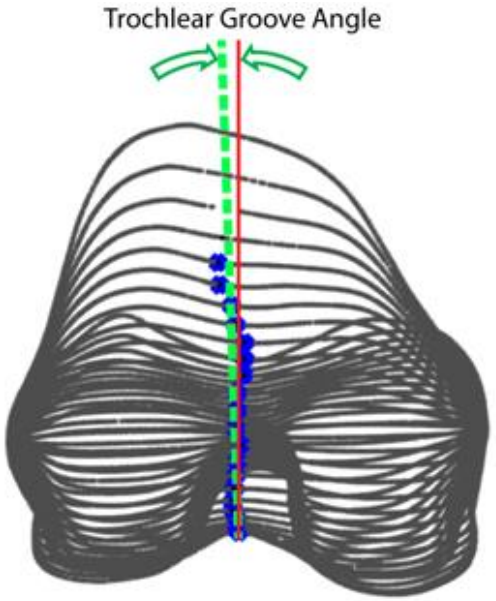
MATLAB

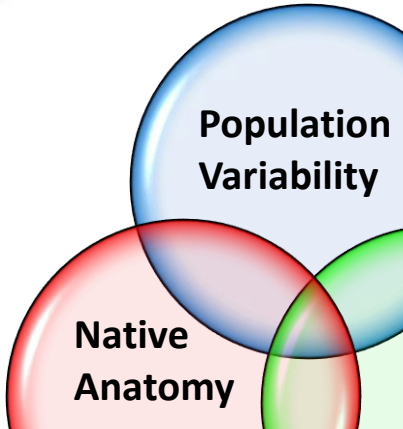


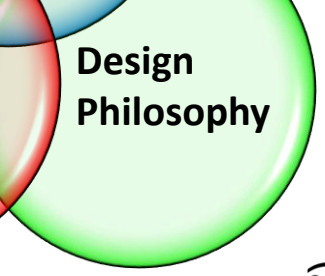
Native Trochlear Orientation

	Trochlear Groove Orientation (°) † Mean ± Standard Deviation [95% range]	
Pooled	-1.4° ± 4.7° [-10.8°, 8.0°]	
Female	-1.0° ± 4.8° [-10.6°, 8.6°]	N.S.
Male	-1.8° ± 4.6° [-11.0°, 7.4°]	
Asian	-2.1° ± 3.9° [-9.9°, 5.7°]	N.S.
Caucasian	-0.6° ± 5.3° [-11.2°, 10.0°]	

† Negative values indicate that the trochlear groove was tilted laterally in distal to proximal direction.







* Data on native femur was illustrated with average and the 95% range.
† Design A was illustrated with the range of allowed groove orientation.
# Data on allowed range of trochlear groove angle not available.

Trochlear Groove Orientation (°)

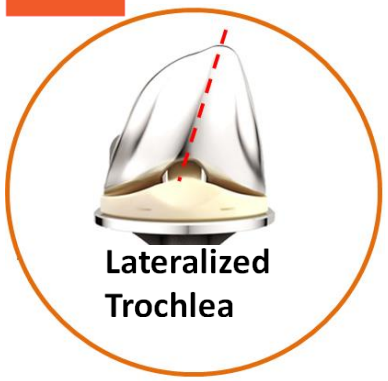
Medial
15
10
5
0
-5
-10
Lateral
-15

Native Femur*

Design A†

Design B
Design C
Design D
Design E

#
#
#
#



Summary of User Experience

- Short script: ~60 lines for this study
- Highly efficient: < 2 minutes per bone for ~30 curves
- Offered easy data I/O based on Python platform
- Projects saved for validation activities, future reference and demo
- Versatile and easy customization of workflow and study variables
- Functions ready to use in a commercially available package