1. Benchmarks

. . . are not a new idea . . .

a set of parameter points in a (your favorite) model (beyond the SM)

- Tool for BSM searches at colliders (past, present, future)
  $\rightarrow$ often it is not feasible to scan over all parameters

- Map out the characteristics of the parameter space

- Take into account all(?) possibilities

- Ensure compatibility with all(?) current bounds
  - searches for new particles
  - (low-energy) flavor bounds
  - (low-energy) electroweak precision bounds
  - cold dark matter
  - . . .

Benchmarks can be used to:

- Study the performance of different detectors
- Study the performance of different experiments
- Perform very detailed studies
- Analyzing the complementarity of different experiments
- Work out synergy effects of different experiments

Prime example from the past: SPS (Snowmass points and slopes)
(especially SPS 1a)
[*hep-ph/0202233*]

External constraints?

If a benchmark is designed to test one sector of a specific model
⇒ should constraints from other sectors be taken into account?
⇒ could they be easily avoided?

If a benchmark is designed to test collider phenomenology

then little changes that do not affect the collider phenomenology can easily avoid:

– bounds from cold dark matter
– bounds on $(g-2)_\mu$
– b physics constraints

My main wish:

Study collider phenomenology in (SUSY) models that are compatible with

– direct experimental searches
– flavor physics constraints
– precision observables constraints

My main wish:

Study collider phenomenology in (SUSY) models that are compatible with
– direct experimental searches
– flavor physics constraints
– precision observables constraints

Special(?) approach for SUSY:

Find/use points as described above (in the (N)MFV MSSM) . . .

that show interesting phenomenology in low- and high-energy experiments

⇒ study the complementarity of the low/high-energy experiments

⇒ study the synergy of the low/high-energy experiments

i.e. combine results from all sources to pin down the (N)MFV MSSM

. . . but this seems to be very difficult

2. Tools

Tools on the market:

- codes for B, K physics observables
- codes for low-energy (ew) observables
- codes for high-energy observables
- codes for the calculation of amplitudes
- codes for connecting the GUT and the (flavor)experimental scale
- codes to pass parameters/results from one code to another
- codes for UT/CKM fits

General questions:

- What is still missing? Are all relevant fields covered?
- How can it be ensured that code/calculation is useful for others?
- Can experimentalists make use of them?
- What are the wishes of the experimentalists?
- Interaction between theory and experiment?

My main question:

One code/tool is good!

Many codes/tools are better!

Q: How can one connect different tools such that

– input/output is compatible
– (combination of) tools can be used by non-experts
  (non-expert = non-author of the code)
  $\Rightarrow$ mostly in the hands of the authors . . .

A: Two obvious possibilities (maybe more?):

1) Interface code that handles input/output $\rightarrow$ SLHA2
2) "master tool": Über-code that takes care

A few words on SLHA2: $\Rightarrow$ MSSM (+ extensions) only!

[*P. Skands et al. '03 - '07*]

SLHA(2) = Collection of rules to unambigously define input/output

– interface for MSSM (+ extensions) tools (new models $\Leftrightarrow$ priv. defs.)
– ASCII format
– Block structure for different parametes/observables
– parameters defined via Lagrangian
– observables defined via "agreement"

Spectrum generators $\to$ cross section/decay packages $\to$ event generators

+ : IT WORKS!
– : only if implemented by the authors of the code
– : "only" for MSSM + extensions

NEW: inclusion of NMFV/RPV/CPV in the MSSM + NMSSM:
SLHA $\to$ SLHA2

I/O made easy via SLHALib2 [*T. Hahn '06*]
C++ classes [*P. Skands '07*]
read/write SLHA2 data, i.e. NMFV/RPV/CPV MSSM, NMSSM

The "master tool"

⇒ effort in collaboration with CMS physicists [*O. Buchmüller et al.*]

Über-code for the combination of different tools:

– tools are included as subroutines

– compatibility ensured by collaboration of authors of "master tool" and authors of "sub tools"

– one "master tool" for one model . . .

⇒ evaluate observables of one parameter point consistently with various tools

Example: flavor observables and high p_T observables
can be combined

A: Two obvious possibilities (maybe more?):

1) Interface code that handles input/output → SLHA2

Enough for flavor?

Flavor specific extension?

More model independent approach?

How to get people converge? (SLHA was a HUGE effort!)

. . . ?

2) "master tool": Über-code that takes care

Wanted/accepted?

How to include more tools?

How to include updates of tools?

. . . ?

3) . . . ?